University of Alberta

MULTIVARIATE PROCESS MODELLING AND MONITORING FOR CHEMICAL PROCESSES

by

Zhengang Han

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

in

Process Control

Department of Chemical and Materials Engineering

Edmonton, Alberta

Spring 2005

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

0-494-08245-3

Your file *Votre référence*
ISBN:
Our file *Notre reference*
ISBN:

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

"Engineering is the art of organizing and directing men and controlling the forces and materials of nature for the benefit of the human race."

- Henry G. Stott, 1907

To

我的父母和尹长宇

my parents and my wife Eunice for their constant love,

support and tolerance

Abstract

One of the most important challenges faced by control engineers is design and implementation of decision support systems that can assist operators to make supervisory control decisions. Operator failing to exercise the appropriate supervisory control decisions often has an adverse effect on product quality, process safety, occupational health and environmental impact. Thus, there exists considerable incentive in developing decision support systems that can provide automated operator assistance for complex plants.

Before design and implementation of decision support systems for process monitoring, one has to obtain a relatively accurate representation of the system under normal operating condition. This step is known as process modelling or system identification. It is widely regarded as a key step towards successful design of process monitoring systems. There are various approaches available in this area such as prediction error method (PEM), subspace identification method (SIM) and multivariate statistical regression method (MSRM). In this thesis, partial least squares (PLS) method was applied onto a bleached chemi-thermomechanical pulp (BCTMP) plant to obtain the process model with various considerations, e.g. nonlinear effects. Further, traditional canonical variate analysis (CVA) was extended to deal with ill-conditioned data via reduced Krylov space and Cayley-Hamilton theorem.

After the process model has been obtained through the above mentioned approaches, one can design process monitoring systems using variety of methods. However, the model quality, i.e. model plant mismatch (MPM), and unknown disturbances can affect process monitoring results significantly. By extending the

Chow-Willsky ([19]) scheme, the effect of process uncertainties can be completely removed for sensor fault detection and diagnosis (FDD). For actuator FDD, the principle components of process uncertainties can also be eliminated so that FDD results are affected in the minimum manner.

A FDD method dealing with multiplicative faults is proposed by using data reconciliation (DR) and gross error detection (GED). The proposed method was then applied to a chemical tank inventory system and successfully identified the location and magnitude of a multiplicative sensor calibration error.

FDD problem under multirate situation is also investigated in this thesis. Using the well-known lifting technique, the multirate discrete time model can be obtained by extending subspace identification method. Once the multirate model is obtained, the structured residual vector approach is used for fault detection and diagnosis. An experimental case study proves the effectiveness of the proposed method.

The fundamental issue of fault detectability is also analyzed in order to provide the bases for FDD research. The general fault detectability problem can be categorized into two subproblems: fault detectability and strong fault detectability. The conditions for both the subproblems are provided and proved.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Sirish L. Shah, for his constant encouragement and patient guidance through all stages of my doctoral work. My special thanks go to him for sparing his invaluable time in helping me make the right decision on several critical occasions. Indeed, my discussions with him were the highlight of the research process for me. His interest in my work, his careful reading and helpful suggestions, as well as his supportive and caring attitude throughout the program, were indispensable to the successful completion of this dissertation.

I would like to thank Profs. Huang, Marquez, Schiavone, Shah, Chen, Hooper, Pedrycz and Nandakumar for teaching me the fundamentals of conducting research. I feel fortunate to have had the opportunity to learn from such wonderful teachers.

I am also grateful to Drs. Shankar Narasimhan and Weihua Li. I learned data reconciliation and gross error detection from Shankar, and had excellent discussions with him regarding the chemical tank inventory project. Weihua introduced me to the subject of fault detection and diagnosis, especially the parity space approach. Their rigorous attitude toward research stimulated me throughout my graduate program.

I also wish to express my gratitude to the Alberta Research Council, Matrikon Inc. and Millar Western Forest Products Ltd. for providing me with opportunities to obtain excellent hands-on experience with industrial problems and application of new techniques in the real world. Without the strong support of Rohit, Johan, George, Chris, Bruce and Hasna, it would have been impossible to take these projects to successful completions.

The computer process control group was a superior environment in which to work. I will always cherish the friendship of my colleagues in this group: Arun, Hari, Shoukat, Weihua, Hancong, Edward, Kannan, Monjur, David, Rumana, Fangwei, Ian, Ruoyu, Yutong, Sadiq, Salim, Syed, Vinay, Bhushan, Amar, Huilan, Folake, Vikas, Raghu, Dongguang, Jianping, Xin, Yimin, Yale, Giti, Ashish, Abhishek and Haitao. Memories of long and enlightening blackboard discussions I had with Weihua, Hari, Shoukat and Xin will stay with me forever.

I would like to acknowledge the Department of Chemical and Materials Engineering, University of Alberta, for providing me the opportunity to pursue my doctoral degree.

I also gratefully acknowledge the financial support I received from the NSERC-Matrikon-ASRA senior industrial research chair program.

Last but not the least, I would like to thank my parents and my wife Eunice for their constant love, support and tolerance. Without them, it would have been impossible for me to accomplish what I have achieved.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Scope of Thesis 4

1.3 Research Objectives 6

1.4 Overview of the Thesis 8

2 Problem Formulation and Process Modelling **11**

2.1 System Description 11

2.1.1 Steady State Processes 11

2.1.2 Dynamic Processes 13

2.1.3 Fault Representation 14

2.2 Tasks of FDD Systems 15

2.3 General Set-up of FDD System 16

2.4 Process Redundancy 17

2.4.1 Physical Redundancy 17

2.4.2 Analytical Redundancy 17

2.5 Process Modelling 18

2.5.1 Prediction Error Method (PEM) 18

2.5.2 Subspace Identification Method (SIM) 20

2.5.3 Multivariate Statistical Regression Method (MSRM) 22

3 Existing FDD Approaches **25**

3.1 Overview 25

3.2 Parity Space Approach 26

3.3 Observer Based Approach 28

3.4 Multivariate Statistical Based Approach 30

3.5 Data Reconciliation (DR) and Gross Error Detection (GED) 33
3.5.1 Global Test (GT) 34
3.5.2 Constraint or Nodal Test (NT) 35
3.5.3 Measurement Test (MT) 35
3.5.4 Generalized Likelihood Ratio (GLR) Test 35

4 Fault Detectability Analysis for Linear Systems 37
4.1 Introduction 37
4.2 Problem Definition 38
4.3 Definitions 39
4.3.1 Parity Equation and Parity Function 39
4.3.2 Residual Generator 40
4.3.3 Fault Detectability and Strong Fault Detectability 43
4.4 Fault Detectability Analysis 46
4.4.1 Condition for Fault Detectability 46
4.4.2 Condition for Strong Fault Detectability 48
4.4.3 Strong Fault Detectability for Output Sensor Faults 50
4.5 Numerical example 52
4.6 Conclusion 53

5 Process Modelling for Pulp Bleaching Process 55
5.1 Introduction 55
5.2 Concept of Softsensor 57
5.3 Systematic Approach for Softsensor Design 58
5.3.1 Modelling Method Selection 60
5.3.2 Quality Indicators 60
5.3.3 Input Variable Selection (IVS) 61
5.4 Process Description 63
5.5 Softsensor Development 67
5.5.1 Objective 67
5.5.2 Preliminary Variable Selection 67
5.5.3 Nonlinear Consideration - Pseudo Varialbes 69
5.5.4 Input Variable Selection 70
5.5.5 Online Implementation and Results 75
5.6 Concluding Remarks 76
5.7 Appendix: the Partial Least Squares (PLS) Method 77

6 Canonical Variate Analysis for Ill-conditioned Data **81**
6.1 Introduction . . . 81
6.2 Numerical Problem of Existing CVA . . . 83
6.2.1 CVA-based Multivariate Regression . . . 83
6.2.2 Numerical Problem with Ill-conditioned Data . . . 85
6.3 CVA Insensitive to Ill-conditioned Data . . . 86
6.3.1 Reduced Krylov Controllability Matrix . . . 86
6.3.2 Optimal Estimation of Parameter Vector . . . 89
6.3.3 Calculation of Latent Variables . . . 89
6.4 Extension to Dynamic Processes . . . 92
6.5 Numerical Examples and Discussions . . . 94
6.5.1 Example 1 . . . 95
6.5.2 Example 2 . . . 97
6.6 Conclusion . . . 99
6.A Derivation of Eqn. 6.32 . . . 102

7 Sensor and Actuator Fault Detection and Diagnosis with the Presence of Process Uncertainties **105**
7.1 Introduction . . . 105
7.2 Problem Formulation . . . 106
7.2.1 System Description . . . 106
7.2.2 Process Uncertainties . . . 107
7.2.3 Problem of FDD in the Presence of Process Uncertainties . . . 108
7.3 Robust Sensor FDD . . . 109
7.3.1 PRV Generation . . . 109
7.3.2 Sensor Fault Detection . . . 110
7.3.3 Sensor Fault Isolation . . . 111
7.3.4 Condition for Perfectly Decoupling the Uncertainty Vector . . . 113
7.4 Robust Actuator FDD . . . 114
7.4.1 Difficulty in Completely Decoupling the Uncertainties from the PRV . . . 114
7.4.2 PRV Generation . . . 115
7.4.3 Conditions for Decoupling the Principal Components of Uncertainties from the PRV and Fault Detectability . . . 116
7.5 Numerical Example and Experimental Case Study . . . 117
7.5.1 Numerical Example . . . 118

7.5.2 Experimental Case Study 126
7.6 Conclusions 127

8 Detection and Diagnosis of Multiplicative Fault by Using Data Reconciliation **131**
8.1 Introduction 131
8.2 Multiplicative Fault Detection and Isolation 132
8.3 Chemical Tank Inventory Process 134
8.4 Monitoring System Design and Diagnosis Results 137
8.4.1 Process Model Description 137
8.4.2 Offline Analysis for Sensor Calibration Error 138
8.4.3 Online Monitoring Design and Implementation 140
8.5 Conclusion 142

9 Fault Detection and Diagnosis with Multirate Data **145**
9.1 Introduction 145
9.2 Problem Formulation 148
9.3 The Lifted Model for the System with NUSM Data 150
9.4 Identification of Residual Models for Fault Detection 152
9.4.1 Description of the Lifted Model with Faults 153
9.4.2 Identification of Γ_s and $\mathbf{H_s}$ 155
9.4.3 Optimal Design of $\mathbf{W}_o$ 158
9.5 An Experimental Case Study 159
9.5.1 The Experimental Pilot Plant 159
9.5.2 Preliminary Work for FDD 159
9.5.3 FDD Results 165
9.6 Concluding remarks 168
9.A Derivation of Eqn. 9.6 168
9.B Derivation of Eqn. 9.10 169
9.C Derivation of Eqn. 9.27 171
9.D Derivation of the algorithm to compute $\mathbf{\Gamma}_s$ 172

10 Concluding Remarks and Future Work **173**
10.1 Concluding Remarks 173
10.2 Recommendations for Future Research 175

List of Figures

1.1 Signal block diagram of the FDD system 5

2.1 Relationships among the three tasks in FDD system 16
2.2 Identification cycle (from [72]). Rectangles: the computer's main responsibility. Ovals: the user's main responsibility 19

5.1 Softsensor development procedure 59
5.2 Block diagram of "stepwise regression" method 64
5.3 Millar Western Bleaching Process 66
5.4 Preliminary results for Brightness versus time in sample periods . . . 69
5.5 Brightness estimation with nonlinear consideration versus time in sampling periods 71
5.6 Stepwise regression results for Brightness 73
5.7 "Optimal" model result for Brightness 74
5.8 Online result for Brightness 75

6.1 A simulated CSTR process 97

7.1 Process schematic of the simulated nonisothermal CSTR system . . . 118
7.2 Step responses from the true model and its identified model of the simulated CSTR process (Solid line: Real system step response; Dashed line: Model step response) 120
7.3 Detection and isolation of a bias fault in the 1^{st} output sensor of the simulated CSTR process using the proposed robust FDD scheme . . 122
7.4 Detection and isolation results of a bias fault in the 1^{st} output sensor of the simulated CSTR process using the Chow–Willsky approach . . 123

7.5 Detection and isolation results of a bias fault in the 1^{st} actuator of the simulated CSTR process using the proposed robust FDD approach . . 125
7.6 Process schematic of the experimental CSTH system 127
7.7 Detection and isolation of a bias fault in the 3^{rd} output sensor of the experimental CSTH system using the proposed robust FDD approach 128
7.8 Detection and isolation results of a bias fault in the 3^{rd} output sensor of the experimental CSTH system using the Chow–Willsky approach 129

8.1 Illustration of different types of calibration errors 133
8.2 Chemical tank inventory process schematics 135
8.3 Mass balance residual using the points without delivery 139
8.4 Residuals when an alarm is generated 143

9.1 Schematic representation of a process where variables are non-uniformly sampled at different rates 146
9.2 Non-uniform and multirate sampling of inputs and outputs 149
9.3 Physical layout of the CSTHS system with the associated hardware . 160
9.4 A sequence of PRVs generated from the validation data 161
9.5 The fault detection index generated from the validation data. The dotted line represents the threshold, 23.209, for the index. 162
9.6 Detection and isolation of a fault in the 1^{st} sensor. The sensitivity of the isolation indices to the fault is [0 1 1 1]. The dotted line in each subplot represents the threshold, 1, for the scaled detection and isolation indices. 166
9.7 Detection and isolation of a fault in the 2^{nd} sensor. The sensitivity of the isolation indices to the fault is [1 0 1 1]. The dotted line in each subplot represents the threshold, 1, for the scaled detection and isolation indices. 167

List of Tables

1.1 Newsworthy incidents in the United States within the year 1994-1995 3

2.1 Brief summary of MSRMs 24

3.1 Incidence matrix 28

5.1 Process variables 68
5.2 Pseudo variables 70
5.3 Input variables selected for the "optimal models": ∘ - selected; × - removed; 72
5.4 Comparisons between the preliminary and the "optimal" model . . . 73

6.1 $\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2$ calculated from 30 samples using various approaches in Example 1 96
6.2 $\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2$ calculated from the remaining 8 samples using various approaches in Example 1 97
6.3 $\|\mathbf{Y} - \hat{\mathbf{Y}}_d\|_F^2$ F calculated from the first 70 data samples using various approaches in Example 2 100
6.4 $\|\mathbf{Y} - \hat{\mathbf{Y}}_d\|_F^2$ F calculated from the remaining data samples using various approaches in Example 2 101

7.1 Incidence matrix to characterize the isolation logic of sensor faults . . 112
7.2 Singular values of matrix $\mathbf{G}_s^\circ$ and their cumulative percentages 124

8.1 Measured variables of chemical tank inventory system 136
8.2 Estimated calibration coefficients 140
8.3 Fault isolation logic 141

9.1 Sensitivity and insensitivity of the 4 SRVs with respect to faulty sensors in the CSTHS system 163

Nomenclature

$'$	Transpose of a matrix
$\boldsymbol{\phi}$	Unmeasured disturbances
$\otimes$	Kronecker tensor product
$\{\mathbf{R_1}, \mathbf{R_2}\}$	Fault model in the state space format
$\dagger$	Moore-Penrose pseudo inverse
$\perp$	Left null space of a matrix
d	Number of process disturbances
$Det(\cdot)$	Determinant of a matrix
$E\{\cdot\}$	Expected value
l	Number of process inputs
m	Number of process outputs
N	Length of process data
n	Process order
p	Number of faults
q	Back shifting operator
$rank(\cdot)$	Rank of a matrix
T_s	Sampling interval

$Vec(\cdot)$ Column vector operator

$\tilde{\mathbf{u}}$ True process inputs

$\tilde{\mathbf{y}}$ True process outputs

$\tilde{\mathbf{z}}$ True process variables including inputs and outputs

$\mathbf{A}(:, m:n)$ The m^{th} to n^{th} columns of $\mathbf{A}$ matrix

$\mathbf{A}(m:n, :)$ The m^{th} to n^{th} rows of matrix $\mathbf{A}$

$\mathbf{e}_i$ The i^{th} column of identity matrix $\mathbf{I}$

$\mathbf{f}$ Fault vector

$\mathbf{f}_u$ Input sensor faults or actuator faults

$\mathbf{f}_y$ Output sensor faults

$\mathbf{f}_z$ General sensor faults

$\mathbf{I}_a$ Identity matrix with dimension a

$\mathbf{K}$ Observer gain

$\mathbf{M}$ Process model in steady state case , see equation (2.1)

$\mathbf{M}^*$ Process model in steady state case , see equation (2.2)

$\mathbf{o}$ Process output measurement noise

$\mathbf{R}_\phi$ Covariance matrix of ϕ

$\mathbf{R}_o$ Covariance matrix of $\mathbf{o}$

$\mathbf{R}_v$ Covariance matrix of $\mathbf{v}$

$\mathbf{u}^*$ Measured process inputs under fault-free case

$\mathbf{v}$ Process input measurement noise

$\mathbf{x}$ Process states

$\mathbf{y}^*$ Measured process outputs under fault-free case

$\mathcal{R}(\cdot)$ Range space of a matrix

$f(\cdot)$ Function of the argument

$\{\mathbf{A_c}, \mathbf{B_c}, \mathbf{C_c}, \mathbf{D_c}, \mathbf{E_c}, \mathbf{J_c}\}$ Continuous process model in state space format

$\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{J}\}$ Discrete process model in state space format

1

Introduction

1.1 Motivation

The filed of process control has made significant advances, in both theory and application, during the last few decades. This progress is achieved largely from the introduction of computer control systems, e.g. distributed control system (DCS) and model predictive control (MPC). *Regulatory control*, which is low-level control, is now routinely managed by a computer system running in an automated manner with little interruption from operators. The level above the regulatory control is called *supervisory control*. At this level, operators must make quick decisions based on complex reasoning about the control of abnormal process situations, start ups and shut downs, optimal control strategies, etc.

However, relying completely on operators to deal with abnormal events and emergencies is increasingly difficult in a complex and highly interconnected process. Modern chemical plants are usually equipped with hundreds of sensors, valves

and complex processing units. Therefore, if even only one critical element of the entire plant malfunctions, the process performance and operation can degrade. The abnormalities have huge impact on process operation and safety. Table 1.1 shows the newsworthy incidents that have been reported due to such faults in the process industry in the United States in a one year period (1994-1995). In these 24 incidents, 12 deaths occurred and hundreds of people were injured. In addition, more than 1 billion dollars were lost with an estimated total impact of 10 billion dollars. Moreover, statistics show that about 70% of industrial accidents are caused by human error. Lack of appropriate tools to help operators deal with abnormal situations is the major cause of human error. It is estimated that the petrochemical industry in the United States incurs approximately 20 billion dollars in losses due to inadequate handling of abnormalities [90]. These statistics provide ample justification for implementing a scheme that can monitor the "health" of all the instruments and processing units.

Process abnormalities can be categorized into two classes according to the seriousness of the problem: *fault* and *failure*. The term *fault* usually refers to those abnormalities that will degrade the efficiency of the entire control system, generate less consistent products or consume more resources, but is not expected to result in catastrophic problems such as an explosion or the shut down of the entire process. In contrast, the term *failure* is used more in the context of the "disasters". To deal with various types of abnormalities, various requirements are needed. In the case of a fault, the requirement is to detect its occurrence with higher sensitivity and with as few false alarms as possible. In addition, the operator must be informed of the possible location and magnitude of the fault. This strategy is referred as *fault detection and diagnosis* (FDD) or *process monitoring*. In the case of a failure, the primary requirement is to predict the failure as soon as possible. However, the distinction between fault and failure can be ambiguous. In some cases, if a process abnormality is not handled appropriately, then a fault can progress to failure, with serious consequences. In this thesis, both fault and failure will be simply referred to as a *fault*.

FDD is an interdisciplinary technology that touches on *modern control theory, system identification, applied engineering statistics, signal processing*, etc. for its theoretical background. In the past three decades, FDD has attracted significant research attention and is viewed as the next major challenge for control engineers ([125]). A typical procedure in FDD consists of two steps: (1) process modelling or system identification; (2) residual generation and evaluation for decision-making. However, in the context of the current status, the procedure of FDD cannot be totally automated in reality. One must include humans in the loop to enable people approve

Table 1.1: Newsworthy incidents in the United States within the year 1994-1995

- Mobil, Torrance, CA explosion and fire, 10/94
- Conoco, Lake Charles, LA, cat cracker fire, 10/94
- Miles Chemical plant, Baytown, TX, acid leak, 11/94
- Koch, Corpus Christi, TX, separator explosion, 11/94
- Mobil, Paulsboro, NJ, chemical releases, 11/94
- Terra Industries, Sioux City, IA, explosion, 12/94
- Chevron, EL Segundo, CA, furnace fire, 1/95
- Mobil, Torrance, CA, gasoline spill, 2/95
- Unocal, San Francisco, CA, acid overflow/leak, 3/95
- Amoco, Cartere, NJ, depot leak/fire, 3/95
- Clark, Blue Island, IL, refinery fire/extended closure, 3/95
- Ultramar, Wilmington, CA, tank leak/fire, 3/95
- Conoco, Ponca City, OK, crude topping unit fire, 3/95
- Sun Oil, Philadelphia, PA, gas leak, 4/95
- Napp Technologies, Lodi, NJ, explosion and fire, 4/95
- Phone-Poulenc, Philadelphia, PA, granulator explosion and fire, 5/95
- Reichhold Chemical, Grundy Co, IL, rupture/fire/spill, 5/95
- BP, Lima and Toledo, OH, refinery fires, 5/95
- Ultramar, Wilmington, CA, crude unit fire, 6/95
- Unocal, San Francisco, CA, naptha tank fire, 6/95
- Tosco, San Francisco, CA, crude unit fire, 6/95
- Murphy Oil, New Orleans, LA, solvent extraction unit fire, 7/95
- Amoco, Texas City, TX, cat cracker explosion and fire, 7/95
- Conoco, Ponca City, OK, refinery fire, 7/95

the final decision through a decision-support system.

The basic purpose of FDD or *process monitoring* is to compare the new measured process variables with the process model obtained under a *fault–free* situation. If there is a large deviation from what the model predicts, the presence of a fault or an event is detected. By further manipulating the deviation, the location of the fault can be determined by comparing it with the *a priori* knowledge of fault signatures.

FDD is closely related to *system identification* or *process modelling*. Identification of a process model under fault-free conditions is always the first step prior to FDD, no matter what approach is used. Without a precise representation of the system, one cannot draw a conclusion about the presence of fault. For most complex chemical processes, it is almost impossible to build a first principle model based on physical/chemical laws. Instead, one can apply system identification techniques to obtain an approximate model of the process. In most cases, even though they are nonlinear in nature, the process dynamics can be approximated by linear models. Throughout this thesis, it is assumed that the process under consideration can be represented by a linear model with acceptable accuracy. Since a majority of the processes are sufficiently well-regulated by linear PID controllers, for the purpose of this thesis, it is reasonable to consider such systems as locally linear.

The general set-up for FDD is shown in Figure 1.1. In this figure, three types of faults are considered: sensor ($\mathbf{f}^y(k)$), actuator ($\mathbf{f}^u(k)$) and process ($\mathbf{f}^p(k)$) faults. The differences between these faults are evident in Figure 1.1, and will be illustrated mathematically in Chapter 2.

1.2 Scope of Thesis

In this thesis, a wide range of process modelling and monitoring approaches are studied, and various problems within this field are tackled. In practice, the design and implementation of a FDD system require two steps:

1. **System identification or process modelling:** Develop a highly accurate model for a process under consideration. The resulting model is usually an empirical one based on input-output data.

2. **Process monitoring or FDD:** Design a scheme that has a high level of sensitivity for the potential fault and a low false alarm rate.

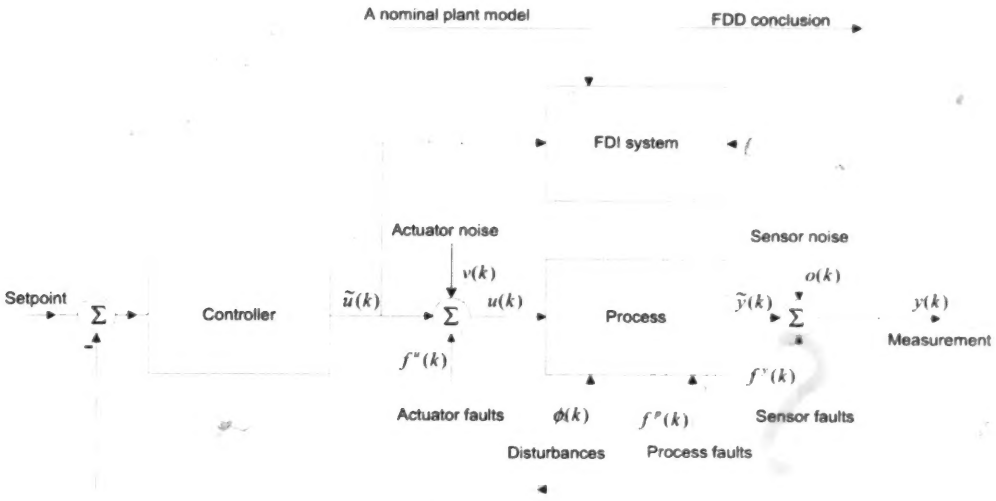

Figure 1.1: Signal block diagram of the FDD system

Thus, this thesis covers two major topics: process modelling and process monitoring. Process modelling is usually considered a prerequisite for process monitoring.

There are several ways of obtaining the process model for a given industrial situation. These include (1) first principle modelling from physical and/or chemical laws, (2) empirical modelling by using input-output data from a designed experiment, and (3) empirical modelling by using input-output data from normal operation. Data collected from industrial settings can be single-rate or multi-rate. The model obtained at this stage must be accurate compared to the normal behavior of the process.

At the process monitoring stage, one must attain a balance between high sensitivity and low false alarm rate. There is a trade-off between these two factors. In statistics, they are termed type I error and type II error respectively. In addition, one must be certain what type of fault is important in a given application. This is a relatively easy task when dealing with sensor/actuator faults as opposed to process faults. The reason for this is that when only sensor/actuator faults are considered, the way they affect the process is apparent. In the case of process faults, one must determine how they are affecting the process, which usually involves considerably more effort.

1.3 Research Objectives

The research objectives of this thesis are described below:

Fault Detectability Analysis

Several fundamental issues of FDD in multivariate linear time-invariant dynamic systems are investigated. The systems are represented by state space models. Then fault detectability and strong fault detectability are defined. By using the parity space method as the universal way of designing residual generators, the conditions of fault detectability and strong fault detectability are provided and proved.

Robust FDD with the Presence of Process Uncertainties

A novel scheme of sensor/actuator fault detection and diagnosis (FDD) is proposed for multivariate dynamic systems in the presence of process uncertainties, including model-plant-mismatch (MPM) and process disturbances. Given an estimated model that can be biased from the true one, the primary residual vector (PRV) for detecting faults in output sensors can be made completely insensitive to process uncertainties under certain conditions. For detecting faults in actuators, the PRV can be made almost insensitive to process uncertainties. Numerical and experimental examples verify the effectiveness of the proposed scheme where comparisons with an existing robust FDD scheme are conducted.

Canonical Variate Analysis for Ill-conditioned Data

Canonical Variate Analysis (CVA) is widely acknowledged as one of the best latent variables-based techniques for multivariate regression. However, existing CVA algorithms will collapse completely if collinearity among the explanatory variables and/or the response variables exist. In such a case, the data are referred to as *ill-conditioned*. Increasing the complexity and instrumentation of industrial processes can result in generation of ill-conditioned data, thereby making the application of CVA difficult. In this thesis, the fundamental weakness of existing CVA is analyzed. Subsequently, a novel CVA algorithm that works for both static and dynamic processes is proposed. The key idea in the newly-developed CVA is to use a truncated Cayley Hamilton series to approximate the inverse of the covariance matrix of the inputs first. Then, the model parameter matrix for a considered process

is estimated by means of a reduced Krylov controllability matrix. Furthermore, latent variables for inputs and outputs are directly calculated from the data and the Krylov matrix without identifying the weight vectors. The newly-developed CVA preserves all advantages of conventional CVA, but is insensitive to ill-conditioned data. Numerical examples and case studies are given to demonstrate the correctness and effectiveness of the theory. Comparisons among the novel CVA, conventional CVA, principal component regression (PCR), and partial least squares (PLS) have also been made.

Subspace Identification for FDD in Systems with Non-uniformly Sampled Multirate Data

To perform FDD in a system with multirate sampled data, existing work assumes the existence of knowledge of a continuous-time (CT) model of the systems under consideration. Then, a discrete-time (DT) model of the system with multirate sampled data is generated by *lifting* the original CT model. Furthermore, based on the DT model, residual models are designed for FDD. This thesis proposes a novel subspace approach to direct identification of a residual model for FDD in a system with non-uniformly sampled multirate (NUSM) data without any knowledge of the considered system. From the identified residual model, an optimal primary residual vector (PRV) is generated for fault detection. Furthermore, by transforming the PRV into a set of structured residual vectors, fault isolation is performed. The proposed algorithms were applied to an experimental pilot plant with NUSM data for sensor FDI, where different types of sensor faults are successfully detected and isolated, fully supporting the correctness and effectiveness of the developed theory.

Industrial Applications

1. *Softsensor development for a thermo-mechanical pulp mill using Partial Least Squares (PLS) approach*

   In the pulp and paper industry, pulp bleaching is essential in order to meet the high quality standards demanded by the market. In this thesis, PLS-based softsensor models are developed for a Bleached Chemi-ThermoMechanical Pulp (BCTMP) process in order to estimate pulp quality variables – brightness, tensile strength, etc. – based on normal and fast sampled process measurements. This thesis provides a systematic softsensor development approach that has

been verified by practical applications. Nonlinear properties of the process are considered from process knowledge and put into the linear framework. In addition, to eliminate inputs that do not have significant effects on the estimated variables, the stepwise regression method is used. The results are in good agreement with the process knowledge. Currently, the developed softsensor models are running successfully online in the bleaching process.

2. *Detection and diagnosis of sensor calibration error in an industrial chemical inventory system.*

   According to one process engineer, "reconciliation of mass balances to monitor chemical tank inventories are a struggle even at the best of times." This thesis addresses the design of offline and online fault detection and diagnosis monitoring system for the caustic tank inventory process at a pulp and paper company. Even with the presence of limited instrument redundancy, the offline monitoring analysis was able to correctly detect and diagnose sensor calibration errors. The scheme is currently undergoing online implementation tests.

1.4 Overview of the Thesis

As outlined in an earlier section of this chapter, this thesis deals with the design and implementation of process modelling and monitoring schemes. To provide a consistent foundation for the process modelling and monitoring problem, Chapter 2 begins with the system description. The general tasks and set-up of FDD systems are reviewed. Thereafter, major existing process modelling techniques are briefly introduced, followed by the key concept of FDD – process redundancy. Chapter 3 reviews existing approaches of FDD, including the parity space, observer-based, multivariate statistical based and data reconciliation/gross error detection approaches.

Chapter 4 discusses fundamental problems in FDD. By defining some commonly-used terminologies (such as residual generator) the issue of fault detectability is categorized into two levels: detectable fault and strongly detectable fault. Thereafter, the criteria for (strong) fault detectability are given and proved. A special case – the output sensor fault detectability – is considered and a simpler condition is achieved.

Chapter 5 presents an industrial example of modelling a complex chemical process in a closed-loop situation by using PLS method. During the development, several factors are considered – including nonlinear transformation and input

variable selection. These considerations improved the performance of the softsensor significantly.

Chapter 6 proposes a novel CVA method that is insensitive to the ill-conditioned data. It is well known that the conventional CVA approach cannot handle data with collinearity. In this chapter, the Cayley-Hamilton theorem and Krylov space are used to obtain an approximation of the inverse of data matrix so that the problem of ill-conditioned data can be resolved. The newly-proposed CVA preserves the orthogonality of latent variables in an optimal manner. The proposed CVA approach is extended to the dynamic modelling case.

Chapter 7 proposes a robust fault detection and isolation method. In this method, process uncertainties – including disturbances and model plant mismatch – can be completely decoupled for the output sensor FDD. When detecting and diagnosing faults in actuators, the method can make the process uncertainties almost decoupled.

Chapter 8 deals with multiplicative faults by using data reconciliation and gross error detection. In contrast to the other chapters, the process considered in this chapter is modelled by mass balance. Thus, the process model is accurate. Then, the least squares optimization method is used to solve the fault isolation problem. The method is verified by an industrial application with satisfactory results.

Chapter 9 discusses the fault detection and diagnosis method in a multirate situation, i.e. different variables are measured at different sampling rates. Beginning with a continuous-time model of the considered system, a discrete-time model with multiple sampling rates can be obtained by using the lifting technique. Furthermore, the residual can be calculated from the the discrete-time model. Thereafter, FDD can be conducted.

The most significant contributions of this thesis are summarized in Chapter 10, and future research directions are suggested.

This thesis has been compiled in a "paper-format" form as defined in the Faculty of Graduate Study and Research guidelines. The different chapters are related as follows:

The modelling aspects of process systems with the end goal of FDD are discussed in Chapters 2, 5 and 6. Chapter 2 provides the introduction to currently used approaches for process modelling. An industrial modelling application, for softsensor based on Partial Least Squares (PLS), is given in Chapter 5. Canonical Variate Analysis (CVA) is another commonly used method for modelling. However, it has a major disadvantage - it cannot handle collinearity in input data set. A novel CVA approach which handles ill-conditioned data is proposed in Chapter 6.

The monitoring aspects of process systems are discussed in Chapters 3, 4, 7 and 9. Chapter 3 provides the introduction to currently used approaches in process monitoring. The fundamental issue of fault detectability is discussed in Chapter 4. Further, a novel FDD approach dealing with process uncertainties is proposed in Chapter 7. It is common in industry to have process inputs and outputs sampled at different rates. Chapter 9 proposes a FDD approach that can handle such data.

Finally, an industrial application concerned with modelling, fault detection and diagnosis for a chemical tank inventory system is given at Chapter 8.

2

Problem Formulation and Process Modelling

2.1 System Description

The systems described in this section are linear. For a nonlinear system, one can linearize the system around the normal operating condition. In most cases, providing that the system is not highly nonlinear, the obtained linearized model can represent the nonlinear process with acceptable accuracy. The systems described are also assumed to be time invariant.

2.1.1 Steady State Processes

In this thesis, the term "steady state process" refers to those processes whose outputs are only affected by its inputs at *one* point in time, i.e. $\mathbf{y}(t) = f(\mathbf{u}(t_d)),\ t_d \leq t$

where $f(\cdot)$ represents a function of the argument. Implicitly, this definition includes processes with pure time delays.

Assume that the normal behavior of a multivariate linear steady state process can be represented by the following equation:

$$\tilde{\mathbf{y}}(t) = \mathbf{M}\tilde{\mathbf{u}}(t_d) + \boldsymbol{\varepsilon}(t) \tag{2.1}$$

where $\tilde{\mathbf{y}}(t) \in \Re^m$ is the true process outputs; $\tilde{\mathbf{u}}(t_d) \in \Re^l$ is the true process inputs with appropriate time delays, $\boldsymbol{\varepsilon}(t)$ is the systematic error involved in the process relationship, and $\mathbf{M} \in \Re^{m \times l}$ is a matrix representing the process relationship. Note that even though Eqn. 2.1 is expressed in the continuous time domain, one can replace t by k to obtain the discrete time domain representation. This can be applied to the other equations in this section as well. With respect to most literatures, the argument t is used to show the variable is a continuous signal, while k denotes the sampled signal.

By further manipulating Eqn. 2.1, one can obtain a simplified model representation:

$$\mathbf{M}^*\tilde{\mathbf{z}}(t) = \boldsymbol{\varepsilon}(t) \tag{2.2}$$

where $\mathbf{M}^* = \mathbf{T}[\mathbf{I}_m \ \ -\mathbf{M}]$ and $\tilde{\mathbf{z}}(t) = [\tilde{\mathbf{y}}'(t) \ \ \tilde{\mathbf{u}}'(t_d)]'$. $\mathbf{T}$ can be any nonsingular matrix with appropriate dimensions. $'$ denotes the transpose of a matrix.

The advantage of formulating a steady state process in the form of Eqn. 2.2 is that one does not need to distinguish all the process variables into "inputs" and "outputs". In the complex industrial environment, categorizing variables as input or output can sometimes be difficult. By using this formulation, one can include all the available process variables in $\tilde{\mathbf{z}}(t)$ without categorizing them. The advantage of this formulation will be evident in Chapter 8.

However, in most cases, the true process variables are not available for data analysis. They are usually measured by sensors, hence, corrupted by measurement noise. Therefore, we have

$$\begin{aligned} \mathbf{y}^*(t) &= \tilde{\mathbf{y}}(t) + \mathbf{o}(t) \\ \mathbf{u}^*(t) &= \tilde{\mathbf{u}}(t) + \mathbf{v}(t) \end{aligned} \tag{2.3}$$

where $\mathbf{y}^*(t)$ and $\mathbf{u}^*(t)$ are measured process outputs and inputs under fault-free conditions; $\mathbf{o}(t)$ and $\mathbf{v}(t)$ are the zero mean normal distributed measurement noises with covariance matrices $\mathbf{R}_o$ and $\mathbf{R}_v$ respectively. Further, it is assumed that $\mathbf{v}(t)$ and $\mathbf{o}(t)$ are mutually independent.

Substituting Eqn. 2.3 for Eqn. 2.2, one can arrive at the following:

$$\mathbf{M}^*\mathbf{z}^*(t) = \mathbf{M}^*\sigma(t) + \boldsymbol{\varepsilon}(t) \tag{2.4}$$

where $\mathbf{z}^*(t) = [(\mathbf{y}^*(t))' \quad (\mathbf{u}^*(t_d))']'$ and $\sigma(t) = [\mathbf{o}'(t) \quad \mathbf{v}'(t_d)]'$. Furthermore, $\mathbf{M}^*\sigma(t)$ follows normal distribution with zero mean and variance $\mathbf{M}^* \begin{bmatrix} \mathbf{R}_o & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_v \end{bmatrix} (\mathbf{M}^*)'$.

2.1.2 Dynamic Processes

In contrast to "steady state process", "dynamic process" refers to those processes whose outputs are NOT only affected by its inputs at *one* time instance. Dynamic processes are very common in reality. Many industrial processes can be better represented by using dynamic processes.

Assume that the normal behavior of a multivariate dynamic process can be represented by the following continuous time linear state space model:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c}\tilde{\mathbf{u}}(t) + \mathbf{E_c}\boldsymbol{\phi}(t) \\ \tilde{\mathbf{y}}(t) &= \mathbf{C_c}\mathbf{x}(t) + \mathbf{D_c}\tilde{\mathbf{u}}(t) + \mathbf{J}_c\boldsymbol{\phi}(t) \end{aligned} \tag{2.5}$$

where $\mathbf{x} \in \Re^n$ is the state, $\boldsymbol{\phi}(t) \in \Re^d$ is the unmeasured disturbance vector ([45]), and $\mathbf{A_c}$, $\mathbf{B_c}$, $\mathbf{C_c}$, $\mathbf{D_c}$, $\mathbf{E_c}$ and $\mathbf{J}_c$ are system matrices with appropriate dimensions. The process is assumed to be observable.

Due to the advantages and popularity of digital control ([4]), one usually converts the process model from continuous to discrete by assuming the process is appropriately sampled. In this section, we assume the process inputs and outputs are sampled at the same rate T_s. The case of multiple sampling rates will be discussed in Chapter 9. From [4], we have the following discrete time dynamic model in forms of state space:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\tilde{\mathbf{u}}(k) + \mathbf{E}\boldsymbol{\phi}(k) \\ \tilde{\mathbf{y}}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\tilde{\mathbf{u}}(k) + \mathbf{J}\boldsymbol{\phi}(k) \end{aligned} \tag{2.6}$$

where $\mathbf{A} = \mathbf{e}^{\mathbf{A_c}T}$, $\mathbf{B} = \int_0^T \mathbf{e}^{\mathbf{A_c}t}\mathbf{B_c}dt$, $\mathbf{C} = \mathbf{C_c}$, $\mathbf{D} = \mathbf{D_c}$, $\mathbf{E} = \int_0^T \mathbf{e}^{\mathbf{A_c}t}\mathbf{E_c}dt$ and $\mathbf{J} = \mathbf{J_c}$.

We consider the case of errors-in-variables (EIV) ([122]), and denote the observed *fault-free* outputs and inputs the same as in Eqn. 2.3.

According to Eqns. 2.5 and 2.6, one can change the general set-up to a specific set-up. For example, removing the items with disturbance ϕ gives a pure deterministic

system. In this thesis, different chapters focus on different aspects of the system and thus deal with different formats of system representation. However, for dynamic systems, Eqns. 2.5 and 2.6 provide the most general expression.

2.1.3 Fault Representation

In this section, we will use the dynamic state space model as an example to identify different types of faults and their effects on the process. These types can also be applied to the steady state case.

Assume that the dynamics of a multivariate process can be represented by the following discrete time linear state space model with faults:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\tilde{\mathbf{u}}(k) + \mathbf{E}\boldsymbol{\phi}(k) + \mathbf{R}_1\mathbf{f}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\tilde{\mathbf{u}}(k) + \mathbf{J}\boldsymbol{\phi}(k) + \mathbf{R}_2\mathbf{f}(k) \end{aligned} \tag{2.7}$$

where $\mathbf{f}(k) \in \Re^p$ is the fault vector and $\{\mathbf{R_1},\ \mathbf{R_2}\}$ are fault models with appropriate dimensions.

It has been proven by Chen and Patton ([15]) that the system shown in Eqn. 2.7 can represent all possible additive faults, including component, parameter, sensor and actuator faults. Different types of faults correspond to different realizations of matrices $\mathbf{R}_1$ and $\mathbf{R}_2$.

Sensor Faults

In a system, the process inputs and outputs are usually measured by sensors, which can be faulty. In Eqn. 2.7, for output sensor faults, $\mathbf{R}_1 = \mathbf{0}$, $\mathbf{R}_2 = \mathbf{I}_m$ and $p = m$. For input sensor faults, $\mathbf{R}_1 = -\mathbf{B}$, $\mathbf{R}_2 = \mathbf{0}$ and $p = l$. Thus, the sensor faults can be illustrated in the following way:

$$\begin{aligned} \mathbf{y}(k) &= \tilde{\mathbf{y}}(k) + \mathbf{f}(k), \quad \text{for output sensor faults} \\ \mathbf{u}(k) &= \tilde{\mathbf{u}}(k) + \mathbf{f}(k), \quad \text{for input senor faults} \end{aligned} \tag{2.8}$$

Throughout this thesis, the symbols $\mathbf{f}_u(k)$ and $\mathbf{f}_y(k)$ will be used for input and output sensor faults respectively. More generally, the sensor faults can be denoted by $\mathbf{f}_z(k)$.

Actuator Faults

A controller's output affects the process through a physical device termed an actuator, which can be malfunctioning for various reasons (for example, normal wear and tear). When the fault happens in actuators, it can be represented by letting $\mathbf{R}_1 = \mathbf{B}$, $\mathbf{R}_2 = \mathbf{0}$ and $p = l$. Actuator faults are also represented by the symbol $\mathbf{f}_u(k)$ in this thesis.

Additive Process Faults

This type of fault can be mathematically represented as an additive item in the state space model (Eqn. 2.7). This fault represents the situation wherein a process running condition has been changed so that the process model is no longer valid. For example, a tank leakage fault can be categorized as this type of fault. In this case, $\mathbf{R}_2 = \mathbf{0}$ and $\mathbf{R}_1$ can be any matrix which represents how the faults affect process dynamics.

Multiplicative Process Faults

Multiplicative process faults are more complex compared to the above-mentioned faults. Multiplicative process faults are difficult to represent in the form of Eqn. 2.7. However, physically, they constitute changes of plant parameters. Such faults best describe the deterioration of plant equipment, such as partial or total loss of power, surface contamination, etc.

2.2 Tasks of FDD Systems

As mentioned above, different types of faults have different properties and affect the system in different ways. This makes the objective of FDD a very challenging one. To simplify the whole problem, one usually divides the issue into several small steps, each of which accomplishes a sub-task of the whole problem. Gertler defines the widely accepted tasks of FDD system in [38], which he calls "three layers of process diagnosis."

Fault detection In this task, the FDD system must indicate whether something in the system is malfunctioning.

Fault isolation The FDD system must determine the location of the faults in this step.

Fault identification The size of the faults will be determined by the FDD system.

Each task listed above is obviously not of equal importance. "Fault detection" provides the foundation for the whole FDD system. One must ensure that the fault can be successfully detected before attempting to determine its location and magnitude (fault isolation and identification). Likewise, without determining the fault location, estimation of fault size is useless to the plant personnel. The sequential relationships among these three tasks is illustrated in Figure 2.1. Fault detection is the most important task and is relatively easy. However, the other two tasks are also very important, and indeed become crucial in instances where a fault can potentially have a catastrophic impact on the plant's operation.

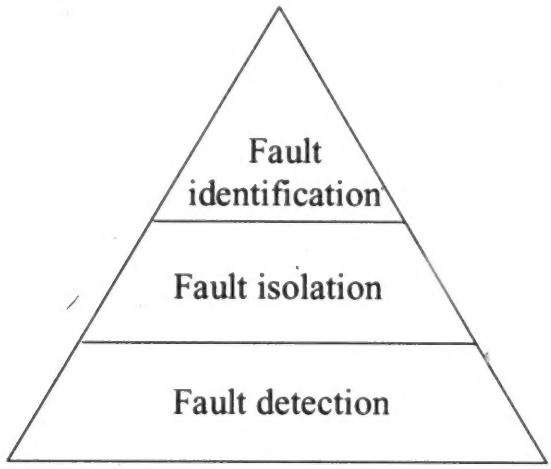

Figure 2.1: Relationships among the three tasks in FDD system

2.3 General Set-up of FDD System

It is usually assumed that $\tilde{\mathbf{u}}(k)$ and $\mathbf{y}(k)$ are available for FDD analysis, because they are controller outputs and the observed process outputs respectively. The detailed process set-up is illustrated in Figure 1.1.

In this section, the elements and knowledge required for FDD system design are discussed in general. Two elements are essential for FDD: process model and process measurement.

1. For process model, one must ensure the process model is accurate enough for FDD. Therefore, when the process measurement is not consistent with these models, one can easily infer that the instruments or the process equipment are abnormal without needing to consider the possibility of model inaccuracy. There are several physical laws that the process must always follow (for example, mass and energy balance). These physical relationships can be used in process modelling and insured to be accurate. However, access to these first principle models is usually a luxury. In the real world, one must live with models with some degree of model-plant-mismatch (MPM). Therefore, there is clearly a need for robust FDD technique.

2. Process measurement must be "redundant". In other words, it is essential that each measured process variable can be inferred from the other measured variables as well as from the process model. This type of redundancy is also referred as "analytical redundancy" ([19]).

2.4 Process Redundancy

2.4.1 Physical Redundancy

Physical redundancy is aimed primarily at detecting and isolating sensor faults. In this case, measurements of the same variable from different sensors are compared. Any serious discrepancy is an indication of the fault of at least one sensor. The measurement that is likely to be correct may be selected in the voting system. Physical redundancy is widely applied in airplanes and space shuttles. Today, it is also applied in some chemical engineering plants. However, physical redundancy has a major disadvantage. One must purchase several identical sensors for one variable, thereby increasing the costs. This disadvantage limits widespread application of physical redundancy.

2.4.2 Analytical Redundancy

Analytical redundancy is achieved from the functional dependence among the process variables and is usually provided by a set of algebraic or temporal relationships among the variables within the system. Analytical redundancy can be classified into two categories ([19], [33]): direct and temporal.

A direct analytical redundancy is associated with algebraic relationships among different sensor measurements. In other words, direct analytical redundancy allows computation of a theoretical sensor value from measurements of other sensors. The computed value can then be compared with the measured value for that sensor. A discrepancy indicates that a sensor fault may have occurred. In contrast, a temporal analytical redundancy is obtained from differential or difference equations among sensors.

The essence of using analytical redundancy in fault diagnosis is to compare the observed system behavior to the process model for consistency. A discrepancy can then be used for fault detection and diagnosis. In this thesis, the focus of research is on how to apply analytical redundancy to FDD.

2.5 Process Modelling

In order to build reliable models, several identification approaches are currently available and widely used. It is worth emphasizing that the modelling methods discussed in this thesis do not include first principle modelling. Because it is based on physical and/or chemical laws, first principle modelling is theoretically the most accurate approach. However, for most complex chemical engineering processes, this approach cannot be easily applied in reality. In this section, I will introduce the prediction error method (PEM), subspace identification method (SIM) and multivariate statistical regression method (MSRM).

As pointed out by Ljung ([72]), the application of process modelling is an interactive procedure involving several trial-and-error loops. The steps involved in process modelling are illustrated in Figure 2.2 (from [72]). The purpose of including Figure 2.2 in this thesis is to emphasize that process modelling is not an automated procedure wherein everything can be accomplished by computers. Human involvement is important in the most successful applications. This will be further illustrated in Chapter 5.

2.5.1 Prediction Error Method (PEM)

The prediction error method (PEM) represents a family of approaches that have the same design criterion. Assume that by giving a set of parameters θ and a model structure, one can obtain an estimate of process outputs $\hat{\mathbf{y}}(k, \theta)$. Then the prediction

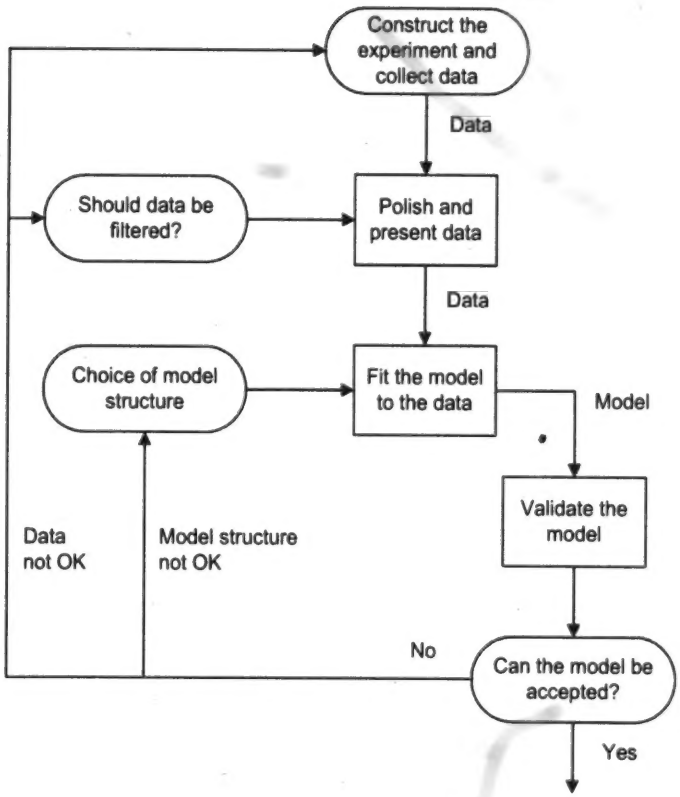

Figure 2.2: Identification cycle (from [72]). Rectangles: the computer's main responsibility. Ovals: the user's main responsibility

error can be derived as:

$$\varepsilon(k,\theta) = \mathbf{y}(k) - \hat{\mathbf{y}}(k,\theta) \tag{2.9}$$

Let the prediction-error sequence be filtered through a stable linear filter $L(q)$:

$$\varepsilon_F(k,\theta) = L(q)\varepsilon(k,\theta) \tag{2.10}$$

where q is the back-shift operator.

Further, from a set of N data, Z^N, and hence of N prediction errors, one can define a function of model parameter θ and process data Z^N:

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N} l(\varepsilon_F(k,\theta)) \tag{2.11}$$

where $l(\cdot)$ is a positive scalar-valued function. The minimization of $V_N(\theta, Z^N)$ with respect to θ then yields the model parameter estimate:

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta, Z^N) \tag{2.12}$$

For the choice of $l(\cdot)$, the quadratic norm is widely used due to the convenience for both computation and analysis:

$$l(\varepsilon) = \frac{1}{2}\varepsilon^2 \tag{2.13}$$

It should be noted that the well-known ordinary least squares (OLS) method is a special case of PEM described above. In addition, PEM covers the maximum likelihood (ML) method and is also closely related to Akaike's information criterion (AIC) ([2]). PEM can be applied to different model structures, such as Auto-Regression with eXogenous variables (ARX), Auto-Regression Moving Average with eXogenous variables (ARMAX), Output Error (OE) and Box-Jenkins (BJ). Refer to Ljung ([72]) for more detail.

2.5.2 Subspace Identification Method (SIM)

The subspace identification method (SIM) has drawn a significant amount of attention in recent years. The most well-known methods in this area are canonical variate analysis (CVA) [67], numerical subspace state space system identification (N4SID) [95], and multivariable output error state space (MOESP) model identification [126].

All of these methods are designed to establish a state space model representation of a dynamic process.

In SIM, first the state sequence of the dynamic system is directly determined from input-output observations, without knowing the model. Thus, an important achievement of the research in subspace identification was to demonstrate how the Kalman filter states can be determined directly from input-output data using linear algebra tools (QR and singular value decompositions) without knowing the mathematical model. Once these states are known, the identification problem becomes a linear least squares problem in the unknown system matrices which can be seen from Eqn. 2.6:

$$\begin{bmatrix} \mathbf{x}(i+1) & \mathbf{x}(i+2) & \cdots & \mathbf{x}(i+j) \\ \tilde{\mathbf{y}}(i) & \tilde{\mathbf{y}}(i+1) & \cdots & \tilde{\mathbf{y}}(i+j-1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}(i) & \mathbf{x}(i+1) & \cdots & \mathbf{x}(i+j-1) \\ \tilde{\mathbf{u}}(i) & \tilde{\mathbf{u}}(i+1) & \cdots & \tilde{\mathbf{u}}(i+j-1) \end{bmatrix} + \begin{bmatrix} \mathbf{E} \\ \mathbf{J} \end{bmatrix} \begin{bmatrix} \phi(i) & \phi(i+1) & \cdots & \phi(i+j-1) \end{bmatrix} \tag{2.14}$$

Even though the state sequence can be determined explicitly, Eqn. 2.14 can be solved "implicitly" as will become clear later without an explicit calculation of the state sequence.

After determining the process order n and the estimated state sequence $\hat{\mathbf{x}}(k)$, one can solve the least squares problem to obtain the state space matrices:

$$\begin{bmatrix} \mathbf{A}_\circ & \mathbf{B}_\circ \\ \mathbf{C}_\circ & \mathbf{D}_\circ \end{bmatrix} = \min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}} \left\| \begin{bmatrix} \hat{\mathbf{x}}(i+1) & \hat{\mathbf{x}}(i+2) & \cdots & \hat{\mathbf{x}}(i+j) \\ \tilde{\mathbf{y}}(i) & \tilde{\mathbf{y}}(i+1) & \cdots & \tilde{\mathbf{y}}(i+j-1) \end{bmatrix} - \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}(i) & \hat{\mathbf{x}}(i+1) & \cdots & \hat{\mathbf{x}}(i+j-1) \\ \tilde{\mathbf{u}}(i) & \tilde{\mathbf{u}}(i+1) & \cdots & \tilde{\mathbf{u}}(i+j-1) \end{bmatrix} \right\|_F^2 \tag{2.15}$$

where $\|\cdot\|_F$ denotes the Frobenius-norm.

By substituting Eqn. 2.3 into Eqn. 2.6 and performing an algebraic manipulation, one can obtain

$$\mathbf{y}_s^*(k) = \mathbf{\Gamma}_s \mathbf{x}(k-s) + \mathbf{H}_s \mathbf{u}_s^*(k) - \mathbf{H}_s \mathbf{v}_s(k) + \mathbf{G}_s \phi_s(k) + \mathbf{o}_s(k) \tag{2.16}$$

where

$$\mathbf{y}_s^*(k) = \begin{bmatrix} \mathbf{y}^*(k-s) \\ \vdots \\ \mathbf{y}^*(k) \end{bmatrix} \in \Re^{m_s}, \quad \mathbf{\Gamma}_s = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^s \end{bmatrix} \in \Re^{m_s \times n},$$

$$\mathbf{H}_s = \begin{bmatrix} \mathbf{D} & & \\ \mathbf{CB} & \mathbf{D} & \\ \vdots & & \ddots \\ \mathbf{CA}^{s-1}\mathbf{B} & & \mathbf{D} \end{bmatrix} \in \Re^{m_s \times l_s}, \quad \mathbf{G}_s = \begin{bmatrix} \mathbf{J} & & \\ \mathbf{CE} & \mathbf{J} & \\ \vdots & & \ddots \\ \mathbf{CA}^{s-1}\mathbf{E} & & \mathbf{J} \end{bmatrix} \in \Re^{m_s \times n_s}$$

Further, $m_s = m(s+1)$, $n_s = n(s+1)$ and $l_s = l(s+1)$. The vectors $\mathbf{o}_s(k) \in \Re^{m_s}$, $\phi_s(k) \in \Re^{n_s}$, and $\mathbf{u}_s^*(k) \in \Re^{l_s}$ have the same formats as $\mathbf{y}_s(k)$. Note that after $\mathbf{\Gamma}_s$ and $\mathbf{H}_s$ are determined, the deterministic system matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ can be determined accordingly. Refer Van Overschee and De Moor ([98]) for details concerning determination of $\mathbf{\Gamma}_s$ and $\mathbf{H}_s$. As they point out, all the SIM can be put into a unified framework, the only difference being that each algorithm uses different weighting matrices.

2.5.3 Multivariate Statistical Regression Method (MSRM)

Differing significantly from PEM and SIM, MSRM was first proposed and applied in the statistics community to deal with the situation of high correlation among measured variables, where the conventional system identification techniques can result in severe numerical difficulties. The main purpose of MSRM is to compress the original variables into fewer *latent variables*, which are independent of each other and sufficient to characterize the information contained in the data.

In general, MSRM relates rows in the process input matrix $\mathbf{U} \in \Re^{N \times l}$ and the process output matrix $\mathbf{Y} \in \Re^{N \times m}$. MSRM assumes that both data matrices $\mathbf{U}$ and $\mathbf{Y}$ are linear functions of a smaller number (a, $a \ll \min(l, m)$) of underlying latent variables:

$$\begin{aligned} \mathbf{U} &= \mathbf{T}\mathbf{P}_u' + \mathbf{E}_u \\ \mathbf{Y} &= \mathbf{T}\mathbf{P}_y' + \mathbf{E}_y \end{aligned} \tag{2.17}$$

where each column of $\mathbf{T} \in \Re^{N \times a}$ is a latent variable defined as a linear combination

of $\mathbf{U}$, that is:

$$\mathbf{T} = \mathbf{UR}, \text{ where } \mathbf{R} \in \Re^{l \times a} \tag{2.18}$$

$\mathbf{P}_u$ and $\mathbf{P}_y$ are the loading matrices for the inputs and outputs respectively, and $\mathbf{E}_u$ and $\mathbf{E}_y$ are error matrices. In reality, complex chemical processes are usually measured extensively, with an enormous number of sensors. However, these measurements are only driven by a limited number of independent variables/directions. Therefore, the space spanned by all the measurements can be similarly represented by a-dimensional space spanned by latent variables. Principal component regression (PCR), partial least squares or projection to latent structures (PLS) and canonical correlation analysis (CCA) are the commonly used MSRMs. They use different methods to identify the latent variables containing an optimal amount of information. Table 2.1 is reproduced from Shi and MacGregor ([120]) to illustrate the differences and similarities among these MSRMs.

MSRM has been used to obtain dynamic models in Auto-Regression with eXogenous variables (ARX) or Finite Impulse Response (FIR) formats. The use of lagged variables in PCA/PCR for dynamic modelling was discussed by Jackson ([56]). The use of lagged variables in PLS was first suggested by Wold *et al.* ([133]) and has been adopted in various applications ([75], [23], [65]).

Table 2.1: Brief summary of MSRMs

	PCR	PLS	CCA
Objective ($i = 1, 2, \cdots, a$)	$\max \mathbf{p}'_{u,i}\mathbf{U}'\mathbf{U}\mathbf{p}_{u,i}$	$\max \mathbf{r}'_i\mathbf{U}'_i\mathbf{Y}\mathbf{p}_{y,i}$ $\mathbf{U}_1 = \mathbf{U}$ $\mathbf{U}_{i+1} = \mathbf{U}_i - \mathbf{t}_i\mathbf{p}'_{u,i}$ $\mathbf{p}_{u,i} = \mathbf{t}'_i\mathbf{U}_i/\mathbf{t}'_i\mathbf{t}_i$	$\max \mathbf{r}'_i\mathbf{U}'\mathbf{Y}\mathbf{p}_{y,i}$
Constraints	$\mathbf{p}'_{u,i}\mathbf{p}_{u,i} = 1$	$\mathbf{r}'_i\mathbf{r}_i = 1$, $\mathbf{p}'_{y,i}\mathbf{p}_{y,i} = 1$	$\mathbf{r}'_i\mathbf{U}'\mathbf{U}\mathbf{r}_i = 1$, $\mathbf{p}'_{y,i}\mathbf{Y}'\mathbf{Y}\mathbf{p}_{y,i} = 1$
LV meaning	Maximize variance of $\mathbf{U}$	Maximize covariance	Maximize correlation
Calculation of LVs	$\mathbf{t}_i = \mathbf{U}\mathbf{p}_{u,i}$	$\mathbf{t}_i = \mathbf{U}\mathbf{r}^*_i$ $\mathbf{r}^*_i = \mathbf{r}'(\mathbf{P}'_u\mathbf{R})^{-1}$	$\mathbf{t}_i = \mathbf{U}\mathbf{r}_i$
Loading vector is eigenvector of matrix	$\mathbf{p}_{u,i}$ $\mathbf{U}'\mathbf{U}$	$\mathbf{r}^*_i$ $\mathbf{U}'_i\mathbf{Y}\mathbf{Y}'\mathbf{U}_i$	$\mathbf{r}_i$ $(\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{Y}(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{U}$
Orthogonality ($i \neq j$)	$\mathbf{p}'_{u,i}\mathbf{p}_{u,j} = \mathbf{0}$ $\mathbf{t}'_i\mathbf{t}_j = \mathbf{0}$	$\mathbf{r}'_i\mathbf{r}_j = \mathbf{0}$ $\mathbf{t}'_i\mathbf{t}_j = \mathbf{0}$	$\mathbf{t}'_i\mathbf{t}_j = \mathbf{0}$
Variance of LV	Generally not unit	Generally not unit	Unit variance
Regression coefficient matrix	$\mathbf{P}_{u,a}(\mathbf{P}'_{u,a}\mathbf{U}'\mathbf{U}\mathbf{P}_{u,a})^{-1}\mathbf{P}'_{u,a}\mathbf{U}'\mathbf{Y}$	$\mathbf{R}^*_a((\mathbf{R}^*_a)'\mathbf{U}'\mathbf{U}\mathbf{R}^*_a)^{-1}(\mathbf{R}^*_a)'\mathbf{U}'\mathbf{Y}$	$\mathbf{R}_a\mathbf{R}'_a\mathbf{U}'\mathbf{Y}$

Note: $\mathbf{t}_i$, $\mathbf{p}_{u,i}$, $\mathbf{r}_i$ and $\mathbf{r}^*_i$ are the a^{th} column vectors of $\mathbf{T}$, $\mathbf{P}_u$, $\mathbf{R}$ and $\mathbf{R}^*$ respectively.

3

Existing FDD Approaches

3.1 Overview

During the past three decades, many theoretical achievements and industrial applications have been made in FDD, and indeed it is still an active field of research. Each year, FDD is one of the active topics in many international conferences organized by IFAC (International Federation of Automatic Control), IEEE (Institute of Electrical and Electronic Engineers), AACC (American Automatic Control Council), and AIChE (American Institute of Chemical Engineers). Numerous prestigious journals in the areas of automatic control, electrical engineering, chemical engineering, mechanical engineering, and aviation publish articles on FDD, and several books are available ([49] [9] [39] [99] [102] [81]).

The pioneering work in FDD was done in the early 1970s. Beard (1971, [12]) and Jones (1973, [59]) used detection filter (a specifically structured observer) to detect and isolate faults. Mehra and Peschon (1971, [84]) first proposed the use of the

Kalman filter for residual generation. Since then, the Kalman filter and various state observers have been used by many researchers in FDD. Early pioneers in FDD also include Deyst and Deckert (1975, [25]), Gustafson *et al.* (1975, [44]), Willsky (1976, [127]), Clark (1978, [21]; 1978, [20]). The first survey paper was published by Willsky (1976, [127]). Later, survey papers were published by Basseville (1983, [6]; 1988, [7]), Isermann (1984, [54]), Gertler (1988, [38], Frank (1990, [33]) and Wise *et al.* (1995, [129]).

3.2 Parity Space Approach

The parity space approach was originally proposed by Chow and Willsky (1984, [19]) to generate residuals for fault detection in a dynamic process represented by state space model (fault isolation is not discussed in their paper). Later, Li and Qin (1999, [105]) developed the structured residuals approach with the maximized sensitivity (SRAMS) approach based on Chow and Willsky's work. Then, Li and Shah (2002, [70]) extended the SRAMS to the structured residual vector based FDD.

By multiplying $(\mathbf{\Gamma}_s^{\perp})'$, the left null space of $\mathbf{\Gamma}_s$, one can obtain the following residual vector from Eqn. 2.16 under fault-free conditions:

$$\begin{aligned} \boldsymbol{\varepsilon}_s^*(k) &= (\mathbf{\Gamma}_s^{\perp})'(\mathbf{y}_s^*(k) - \mathbf{H}_s\mathbf{u}_s^*(k)) \\ &= (\mathbf{\Gamma}_s^{\perp})'(-\mathbf{H}_s\mathbf{v}_s(k) + \mathbf{G}_s\boldsymbol{\phi}_s(k) + \mathbf{o}_s(k)) \end{aligned} \tag{3.1}$$

where on the right hand side, the first line is the computational form of the residual $\boldsymbol{\varepsilon}_s^*(k)$; and the second line is the internal form showing what the residual contains under fault-free situation.

Under the assumption that the process is observable, $rank(\mathbf{\Gamma}_s) = n,\ \forall\ s \geq n$, where $rank(\cdot)$ denotes the rank of a matrix. Therefore, $rank(\mathbf{\Gamma}_s^{\perp}) = m_s - n$, meaning the residual $\boldsymbol{\varepsilon}_s^*(k)$ has $m_s - n$ independent elements.

In Eqn. 3.1, $\boldsymbol{\varepsilon}_s^*(k)$ is only a moving average of measurement noises and disturbances. Under the assumption that $\mathbf{v}(k)$, $\mathbf{o}(k)$ and $\boldsymbol{\phi}(k)$ are Gaussian-distributed random vectors with zero mean and the process is fault-free, $\boldsymbol{\varepsilon}_s^*(k)$ is still a zero mean Gaussian-distributed random vector, i.e. $\boldsymbol{\varepsilon}_s^*(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_\varepsilon)$ where $\mathbf{R}_\varepsilon$ is the covariance matrix of $\boldsymbol{\varepsilon}_s^*(k)$.

Under the fault situation, by using the representation in Eqn. 2.7, the residual can

be shown in the following:

$$\begin{aligned}\varepsilon_s(k) &= (\mathbf{\Gamma}_s^{\perp})'(\mathbf{y}_s(k) - \mathbf{H}_s\mathbf{u}_s(k)) \\ &= (\mathbf{\Gamma}_s^{\perp})'\mathcal{H}_s\mathbf{f}_s(k) + (\mathbf{\Gamma}_s^{\perp})'(-\mathbf{H}_s\mathbf{v}_s(k) + \mathbf{G}_s\phi_s(k) + \mathbf{o}_s(k)) \\ &= (\mathbf{\Gamma}_s^{\perp})'\mathcal{H}_s\mathbf{f}_s(k) + \varepsilon_s^*(k)\end{aligned} \tag{3.2}$$

where

$$\mathcal{H}_s = \begin{bmatrix} \mathbf{R}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CR}_1 & \mathbf{R}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{s-1}\mathbf{R}_1 & \mathbf{CA}^{s-2}\mathbf{R}_1 & \cdots & \mathbf{R}_2 \end{bmatrix} \in \Re^{m_s \times p_s}$$

Due to the deterministic property of the fault signal, the residual $\varepsilon_s(k)$ is no longer a zero mean random vector, i.e. $\varepsilon_s(k) \sim \mathcal{N}\left((\mathbf{\Gamma}_s^{\perp})'\mathcal{H}_s\mathbf{f}_s(k), \mathbf{R}_\varepsilon\right)$. Therefore, the mean of vector $\varepsilon_s(k)$ is only affected by the fault signal $\mathbf{f}_s(k)$. $\varepsilon_s(k)$ is referred as a primary residual vector (PRV). Based on this fact, we define the following Squared Weighted Residual (SWR)

$$SWR(k) = \varepsilon_s'(k)\mathbf{R}_\varepsilon^{-1}\varepsilon_s(k)$$

for fault detection, which follows a chi-square distribution with a degree of freedom $m_s - n$ under the fault-free situation, i.e. $SWR(k) \sim \chi^2(m_s - n)$. Thus, a fault can be detected by observing whether $SWR(k)$ exceeds a pre-determined confidence limit.

In order to further isolate faults, the primary residual vector $\varepsilon_s(k)$ can be transformed into a set of structured residual vectors (SRVs) as following:

$$\mathbf{r}_i(k) = \mathbf{W}_i\varepsilon_s(k) = \mathbf{W}_i(\mathbf{\Gamma}_s^{\perp})'\mathcal{H}_s\mathbf{f}_s(k) + \mathbf{W}_i\varepsilon_s^*(k) \tag{3.3}$$

The original concept of structured residuals comes from Gertler and his co-workers (1985, [36]; 1990, [37]). It was then extended to structured residual approach with maximum sensitivity (SRAMS) by Qin and Li (1999, [105]). Further the structured residual vector (SRV) based approach is proposed by Li and Shah (2002, [70]).

In order to design the matrices $\mathbf{W}_i$, one needs to determine the structure characterized by an incidence matrix for $\mathbf{r}_i(k)$. An incidence matrix consists of binary codes "0" and "1". The rows of incidence matrix correspond to different $\mathbf{r}_i(k)$, while the columns correspond to different fault sources. Table 3.1 gives an example of designing an incidence matrix. Assuming that only a single fault can happen at each time, one can design an incidence matrix as depicted in Table 3.1.

Table 3.1: Incidence matrix

	$f_1(k)$	$f_2(k)$	$\cdots$	$f_p(k)$
$\mathbf{r}_1(k)$	0	1		1
$\mathbf{r}_2(k)$	1	0		1
$\vdots$			$\ddots$	
$\mathbf{r}_p(k)$	1	1		0

In Table 3.1, "0" indicates the corresponding SRV is not affected by the corresponding fault and "1" denotes the SRV has maximum correlation with the fault. Therefore, by observing the behavior of the SRVs, one can correctly pinpoint the fault source. For example, if $\mathbf{r}_1(k)$ is not affected by a fault while all the other SRVs are, it can be concluded that $f_1(k)$ is happening in the process. Interested readers can refer to [70] for the details.

3.3 Observer Based Approach

As indicated in Eqn. 2.6, the observer based approach uses the following architecture to generate the residual:

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{K}(\mathbf{y}(k) - \hat{\mathbf{y}}(k)) \\ \hat{\mathbf{y}}(k) &= \mathbf{C}\hat{\mathbf{x}}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{3.4}$$

By introducing Eqn. 3.4 into Eqn. 2.7, one can obtain the following:

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= (\mathbf{A} - \mathbf{K}\mathbf{C})\tilde{\mathbf{x}}(k) + (\mathbf{R}_1 - \mathbf{K}\mathbf{R}_2)\mathbf{f}(k) + (\mathbf{E} - \mathbf{K}\mathbf{J})\phi(k) \\ \mathbf{e}(k) &= \mathbf{C}\tilde{\mathbf{x}}(k) + \mathbf{R}_2\mathbf{f}(k) + \mathbf{J}\phi(k) \end{aligned} \tag{3.5}$$

where $\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$ and $\mathbf{e}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$.

Therefore, the residual generated for FDD can be represented as

$$\mathbf{r}(k) = \mathbf{Q}\mathbf{e}(k) \tag{3.6}$$

where $\mathbf{Q}$ is the transformation matrix.

In the observer based approach, there are two design parameters: $\mathbf{K}$ and $\mathbf{Q}$. The different designs within this category distinguish themselves on the basis of the different selection of these two design parameters. In order to illustrate the idea of the observer based method, Keller's approach (1999, [61]) is introduced here.

Assume the process can be represented by the following discrete time state space equation:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{R}_1\mathbf{f}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \tag{3.7}$$

where the noise $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are zero mean uncorrelated random signals with covariance $\begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$.

We have the following definitions from Liu and Si (1997, [71]):

Definition 3.1 *The linear time invariant system 3.7 is said to have fault detectability indices* $\rho = \{\rho_1, \rho_2, \cdots, \rho_p\}$ *if* $\rho_i = \min\{v : \mathbf{C}\mathbf{A}^{v-1}\mathbf{r_i} \neq \mathbf{0}, v = 1, 2, \cdots\}$ *where* $\mathbf{R_1} = \begin{bmatrix} \mathbf{r_1} & \cdots & \mathbf{r_p} \end{bmatrix}$.

Definition 3.2 *Assume that the system has finite fault detectability indices. The fault detectability matrix* $\mathbf{L}$ *is defined as* $\mathbf{L} = \mathbf{C}\mathbf{\Psi}$ *with* $\mathbf{\Psi} = [\mathbf{A}^{\rho_1-1}\mathbf{r_1} \;\; \mathbf{A}^{\rho_2-1}\mathbf{r_2} \;\; \cdots \;\; \mathbf{A}^{\rho_p-1}\mathbf{r_p}]$.

Denote $b = \max\{\rho_i, i = 1, 2, \cdots, p\}$ as the maximum value of fault detectability indices. Accordingly, $\mathbf{f}(k)$ and $\mathbf{R}_1$ can be rearranged to $\bar{\mathbf{f}}(k) = [\bar{\mathbf{f}}_1'(k) \;\; \bar{\mathbf{f}}_2'(k) \;\; \cdots \;\; \bar{\mathbf{f}}_b'(k)]'$ and $\bar{\mathbf{R}}_1 = [\bar{\mathbf{R}}_1^1 \;\; \bar{\mathbf{R}}_1^2 \;\; \cdots \;\; \bar{\mathbf{R}}_1^b]$, where $\bar{\mathbf{f}}_i(k)$ represents the elements in $\bar{\mathbf{f}}(k)$ which have detectability index ρ_i. Therefore, the system shown in Eqn. 3.7 can be rewritten as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \bar{\mathbf{R}}_1\bar{\mathbf{f}}(k) + \mathbf{w}_k \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}_k \end{aligned} \tag{3.8}$$

Consequently, the detectability matrix is then given by:

$$\bar{\mathbf{L}} = \mathbf{C}\bar{\mathbf{\Psi}}, \quad \bar{\mathbf{\Psi}} = [\bar{\mathbf{R}}_1^1 \;\; \mathbf{A}\bar{\mathbf{R}}_1^2 \;\; \cdots \;\; \mathbf{A}^{b-1}\bar{\mathbf{R}}_1^b] \tag{3.9}$$

An observer for fault detection and diagnosis can be mathematically expressed as:

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{K}(k)(\mathbf{y}(k) - \hat{\mathbf{y}}(k)) \\ \hat{\mathbf{y}}(k) &= \mathbf{C}\hat{\mathbf{x}}(k) \end{aligned} \tag{3.10}$$

where $\hat{\mathbf{x}}(k)$ and $\hat{\mathbf{y}}(k)$ denote the state and output estimation vectors. From Eqns. 3.9 and 3.10, the state estimation error $\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$ and the output residual $\mathbf{e}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$ have the following relationship:

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= (\mathbf{A} - \mathbf{K}(k)\mathbf{C})\tilde{\mathbf{x}}(k) + \bar{\mathbf{R}}_1\bar{\mathbf{f}}(k) - \mathbf{K}(k)\mathbf{v}(k) + \mathbf{w}(k) \\ \mathbf{e}(k) &= \mathbf{C}\tilde{\mathbf{x}}(k) + \mathbf{v}(k) \end{aligned} \tag{3.11}$$

The output residual $\mathbf{e}(k)$ can be used for fault detection purposes. The problem is how to design the observer gain matrix $\mathbf{K}(k)$. The following theorem provides the solution.

Theorem 3.1 *(From [61]) Under the assumption of rank$(\bar{\mathbf{L}}) = p$, solution of the algebraic constraint $(\mathbf{A} - \mathbf{K}(k)\mathbf{C})\bar{\mathbf{\Psi}} = \mathbf{0}$ can be parameterized as*

$$\mathbf{K}(k) = \boldsymbol{\omega}\mathbf{Q} + \bar{\mathbf{K}}(k)\mathbf{\Sigma} \tag{3.12}$$

with $\mathbf{\Sigma} = \alpha(\mathbf{I}_m - \bar{\mathbf{L}}\mathbf{Q})$, $\mathbf{Q} = \bar{\mathbf{L}}^{\dagger}$ and $\boldsymbol{\omega} = \mathbf{A}\bar{\mathbf{\Psi}}$, where $\bar{\mathbf{K}}(k) \in \Re^{n\times(m-p)}$ is the reduced gain describing the remaining design of freedom, $\bar{\mathbf{L}}^{\dagger}$ is the generalized inverse or pseudo-inverse of $\bar{\mathbf{L}}$ and $\alpha \in \Re^{(m-p)\times m}$ is an arbitrary matrix determined so that matrix $\mathbf{\Sigma}$ is of full rows rank.

At this stage, one must determine the reduced gain $\bar{\mathbf{K}}(k)$. This reduced gain $\bar{\mathbf{K}}(k)$ can be designed by minimizing the trace of the estimation error covariance matrix, which is consistent with Kalman filter theory. The following theorem solves the problem of designing $\bar{\mathbf{K}}(k)$:

Theorem 3.2 *(From [61]) The fault detection filter can be described by the following relations:*

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \omega\mathbf{r}(k) + \bar{\mathbf{K}}(k)\bar{\mathbf{r}}(k)$$

$$\bar{\mathbf{P}}(k+1) = (\bar{\mathbf{A}} - \bar{\mathbf{K}}(k)\bar{\mathbf{C}})\bar{\mathbf{P}}(k)(\bar{\mathbf{A}} - \bar{\mathbf{K}}(k)\bar{\mathbf{C}})' + \bar{\mathbf{K}}(k)\bar{\mathbf{V}}\bar{\mathbf{K}}'(k) + \bar{\mathbf{W}}$$

$$\bar{\mathbf{K}}(k) = \bar{\mathbf{A}}\bar{\mathbf{P}}(k)\bar{\mathbf{C}}'(\bar{\mathbf{C}}\bar{\mathbf{P}}(k)\bar{\mathbf{C}}' + \bar{\mathbf{V}})^{-1}$$

where $\bar{\mathbf{A}} = \mathbf{A} - \omega\mathbf{Q}\mathbf{C}$, $\bar{\mathbf{C}} = \Sigma\mathbf{C}$, $\bar{\mathbf{V}} = \Sigma\Sigma'$, $\bar{\mathbf{W}} = \mathbf{W} + \omega\mathbf{Q}\mathbf{Q}'\omega'$, $\bar{\mathbf{r}}(k) = \Sigma\mathbf{e}(k)$ and $\mathbf{r}(k) = \mathbf{Q}\mathbf{e}(k)$. Therefore, $\mathbf{r}(k) = \mathbf{Q}\mathbf{e}^{*}(k) + [\bar{\mathbf{f}}_1'(k-1) \;\; \bar{\mathbf{f}}_2'(k-2) \;\; \cdots \;\; \bar{\mathbf{f}}_b'(k-b)]'$, each element of which can only be affected by one fault. In addition, $\mathbf{r}(k)$ can also be regarded as a stochastic deadbeat observer of the fault magnitude.

3.4 Multivariate Statistical Based Approach

The ideas of PCA and PLS were first introduced by Pearson (1901, [103]) and Wold (1966, [131]) respectively. The use of PCA/PLS to build reduced dimension models for the purpose of process monitoring and fault diagnosis has been well researched in recent years ([63] [130] [56] [134]). Since the approaches in this category are similar to each other, the PCA is selected to illustrate the concept of multivariate statistical based approach.

The purpose of PCA is to explain the covariance structure through a set of orthogonal coordinates which are the combination of original variables in the reduced dimension space. Consider a properly scaled data matrix $\mathbf{X} \in \Re^{N \times \gamma}$ where N is the number of samples and $\gamma = m + l$ is the number of variables including inputs and outputs. Assume that all the variables are scaled to be zero mean and unit variance.

In PCA, the first principal component (PC) $\mathbf{t}_1 = \mathbf{X}\mathbf{p}_1$, where $\mathbf{p}_1$ is the coefficient vector for $\mathbf{t}_1$, is a linear combination of original process variables that accounts for the maximum variance of the data matrix. Mathematically, this optimization problem can be expressed as follows:

$$\begin{aligned} &\max \ cov(\mathbf{t}_1) = \max_{\mathbf{p}_1} \mathbf{t}_1'\mathbf{t}_1 = \mathbf{p}_1'\mathbf{X}'\mathbf{X}\mathbf{p}_1 \\ &s.t. \ \ \mathbf{p}_1'\mathbf{p}_1 = 1 \end{aligned} \tag{3.13}$$

The constraint in Eqn. 3.13 is to avoid an infinite $\mathbf{p}_1$ to maximize the objective function. The solution of the above constrained optimization problem can be easily obtained by using the *Lagrangian multiplier*:

$$\mathbf{p}_1 = \text{the eigenvector associated with the largest eigenvalue of matrix } \mathbf{X}'\mathbf{X} \tag{3.14}$$

Therefore, the information unexplained by the first principal component (PC) in the data matrix is

$$\mathbf{E}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1' \tag{3.15}$$

where $\mathbf{E}_1 \in \Re^{N \times \gamma}$ is called the residual matrix. The next step is to find the second PC to explain the maximum variability in matrix $\mathbf{E}_1$. The objective function is similar to Eqn. 3.13, but with the added constraint $\mathbf{p}_1'\mathbf{p}_2 = \mathbf{0}$, which ensures $\mathbf{p}_1$ and $\mathbf{p}_2$ are orthogonal. From this new optimization problem, the second loading vector $\mathbf{p}_2$ can be obtained:

$$\begin{aligned} \mathbf{p}_2 \ = \ &\text{the eigenvector associated with the second} \\ &\text{largest eigenvalue of matrix } \mathbf{X}'\mathbf{X} \end{aligned} \tag{3.16}$$

Similarly, γ PCs can be obtained in order to capture all the information in the matrix. However, in most cases, due to the redundancy and noise in the data, n_a ($n_a \ll \gamma$) PCs can capture most of the variability in $\mathbf{X}$, and $\mathbf{X}$ can be expressed in the following way:

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1' + \mathbf{t}_2\mathbf{p}_2' + \cdots + \mathbf{t}_{n_a}\mathbf{p}_{n_a}' + \mathbf{E} = \hat{\mathbf{X}} + \mathbf{E} \tag{3.17}$$

where $\hat{\mathbf{X}} = \sum_{i=1}^{n_a} \mathbf{t}_i \mathbf{p}_i'$ and $\mathbf{E} = \sum_{i=n_a+1}^{\gamma} \mathbf{t}_i \mathbf{p}_i'$.

Ideally, n_a is chosen in a way such that there is no significant process information left in $\mathbf{E}$, where $\mathbf{E}$ represents the random error. PCA models are formed by retaining only the loading vectors $\mathbf{p}_i$ $(i = 1, \cdots, n_a)$ that describe the systematic variations in the data. It is important to note that properly scaling the original data is crucial in PCA. If the variables are not properly scaled, some trivial variables may dominate the variance-covariance information in matrix $\mathbf{X}$ due to the measurement units.

After a PCA model based on normal operating data is obtained using above algorithm, it can be utilized for the purpose of FDD. Denote $\mathbf{x}_{new} \in \Re^{1\times\gamma}$ as the new observation from the process. Projecting it on to the latent PCA space gives the scores $\mathbf{t}_{new} \in \Re^{1\times n_a}$:

$$\mathbf{t}_{new} = \mathbf{x}_{new}\mathbf{P} \tag{3.18}$$

where $\mathbf{P} = [\mathbf{p}_1 \ \ \mathbf{p}_2 \ \ \cdots \ \ \mathbf{p}_{n_a}]$.

Therefore, the model residual is:

$$\mathbf{e}_{new} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new} = \mathbf{x}_{new} - \mathbf{t}_{new}\mathbf{P}' = \mathbf{x}_{new} - \mathbf{x}_{new}\mathbf{P}\mathbf{P}' = \mathbf{x}_{new}(\mathbf{I} - \mathbf{P}\mathbf{P}') \tag{3.19}$$

Hotelling T^2 and Q statistics (Sum of Prediction Error or SPE) are used to monitor the process. The T^2 statistic based on the first n_a PCs is defined as:

$$T^2 = \sum_{i=1}^{n_a} \frac{\mathbf{t}_i^2}{\lambda_i} \tag{3.20}$$

where λ_i is the corresponding eigenvalue of the covariance matrix of $\mathbf{X}$. The confidence limit of T^2 at confidence level $(1-\alpha)$ is related to the F–distribution as follows:

$$T^2_{\gamma,n_a} = \frac{(\gamma-1)n_a}{\gamma - n_a} F^{\alpha}(n_a, \gamma - n_a) \tag{3.21}$$

where $F^{\alpha}(n_a, \gamma - n_a)$ is the upper $100\alpha\%$ critical point of the F distribution with n_a and $\gamma - n_a$ degrees of freedom ([124]). The confidence limit of T^2 gives an ellipsoid in a n_a dimensional space. $100\alpha\%$ of the chance, the new score should be within this ellipsoid under fault-free situation.

However, variations in the process could be associated with the breakdown of the correlation structure among the measured variables while still within the confidence limit of T^2. Therefore, monitoring the process only with T^2 is not sufficient. The SPE chart is also used in conjunction with the T^2 chart. The SPE is defined as

$$SPE = \mathbf{e}\mathbf{e}' \tag{3.22}$$

where $\mathbf{e} \in \Re^{1\times\gamma}$ is the model residual. Jackson and Mudholkar (1979, [55]) provide the confidence limit of SPE:

$$Q_\alpha = \Theta_1 \left[1 + \frac{c_\alpha h_0 \sqrt{2\Theta_2}}{\Theta_1} + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1^2}\right]^{\frac{1}{h_0}} \tag{3.23}$$

where $\Theta_i = \sum_{j=n_a+1}^{n} \lambda_j^i,\ \forall\, i = 1,2,3,\ h_0 = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2^2}$ and c_α is the confidence limits for the $1-\alpha$ percentile in a normal distribution. When the SPE value based on a new observation is beyond the confidence limit, an abnormal event is detected.

On the other hand, in order to isolate the source of the abnormal event, fractional contribution of each process variables can be used:

$$\alpha_i = \frac{SPE_i}{SPE},\quad \forall\, i = 1,2,\cdots,\gamma \tag{3.24}$$

where SPE_i denotes the square of the i^{th} element of the error vector $\mathbf{e}$. If the contribution from some process variables is significant, then these variables are most likely to be the cause of the abnormality ([85]). Although the contribution plots cannot unequivocally diagnose the cause, they do provide much greater insight into possible cause and thereby greatly narrow the search ([76]).

3.5 Data Reconciliation (DR) and Gross Error Detection (GED)

Based on Eqn. 2.3, the data reconciliation issue can be stated as the following constrained optimization problem:

$$\begin{aligned} \min_{\tilde{\mathbf{z}}} \quad & (\mathbf{z}^* - \tilde{\mathbf{z}})' \mathbf{R}_\sigma^{-1} (\mathbf{z}^* - \tilde{\mathbf{z}}) \\ \text{s.t.} \quad & \\ & \mathbf{M}^* \tilde{\mathbf{z}} = \mathbf{0} \end{aligned} \tag{3.25}$$

If it is assumed that the measurement noises are normally distributed with covariance matrix $\mathbf{R}_\sigma$, the resolution of Eqn. 3.25 gives maximum likelihood estimates of process variables.

The estimate of the process variables, $\hat{\tilde{\mathbf{z}}}$, can be determined as follows:

$$\hat{\tilde{\mathbf{z}}} = \mathbf{z}^* - \mathbf{R}_\sigma (\mathbf{M}^*)' (\mathbf{M}^* \mathbf{R}_\sigma (\mathbf{M}^*)')^{-1} \mathbf{M}^* \mathbf{z}^* \tag{3.26}$$

With nonlinear constraints (in which case the process model must be represented in nonlinear format), data reconciliation can also be conducted by introducing the nonlinear programming techniques in solving the weighted least squares problem with nonlinear constraints.

Based on data reconciliation, the technique of gross error detection (GED) is introduced to handle the case of non-Gaussian distributed errors in the measurements, such as sensor malfunction or process leaks. These errors are usually referred to as gross error. The presence of gross errors violates the statistical basis of data reconciliation. Thus, one must check for the presence of gross errors in the measurement data before performing DR.

Several works in the literature have attempted to deal with the problem of the location of gross errors. Mah *et al.* (1976, [78]) developed a series of rules based on graph-theoretical results. A serial elimination algorithm was first proposed by Ripps (1965, [110]) and extended by Nogita (1972, [91]). A more systematic approach was developed by Romagnoli and Stephanopoulos (1981, [112]) and Romagnoli (1983, [111]) to analyze a set of measurement data in the presence of gross errors. Survey papers of the available methodologies are given by Mah (1990, [79]) and Crowe (1996, [22]).

Denote the residual to be:

$$\mathbf{r}(k) = \mathbf{M}^{*}\mathbf{z}(k) \tag{3.27}$$

If the process variable $\mathbf{z}(k)$ is only affected by the measurement noise $\sigma(k)$, the residual $\mathbf{r}(k)$ follows a multivariate normal distribution with zero mean and covariance matrix $\mathbf{R}_r = \mathbf{M}^{*}\mathbf{R}_\sigma(\mathbf{M}^{*})'$, where $\mathbf{R}_\sigma$ is the covariance matrix of $\sigma(k)$, i.e. $\mathbf{r}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_r)$. However, in the presence of gross errors, the residual $\mathbf{r}(k)$ does not follow the normal distribution. Therefore, one can conduct statistical tests to determine the presence of gross errors.

3.5.1 Global Test (GT)

The global test ([110], [3], [77]) uses SWR as the test statistic given by:

$$\eta(k) = \mathbf{r}'(k)\mathbf{R}_r^{-1}\mathbf{r}(k) \tag{3.28}$$

Under fault-free conditions, $\eta(k)$ follows a central chi-square distribution, i.e. $\eta(k) \sim \chi^2(rank(\mathbf{M}^{*}))$. Therefore, with a pre-selected significance level α, one can determine the critical value $\chi^2_\alpha(rank(\mathbf{M}^{*}))$. If $\eta(k) > \chi^2_\alpha(rank(\mathbf{M}^{*}))$, a gross error is detected.

3.5.2 Constraint or Nodal Test (NT)

The Constraint or Nodal test ([108], [78]) define the test statistics as follows:

$$\eta_i = \frac{|r_i|}{\sqrt{\mathbf{R}_r(i,i)}} \tag{3.29}$$

where r_i represents the i^{th} element of vector $\mathbf{r}(k)$ and $\mathbf{R}_r(i,i)$ denotes the i^{th} diagonal element of matrix $\mathbf{R}_r$. It can be proved that η_i follows a standard normal distribution, i.e. $\eta_i \sim \mathcal{N}(0,1)$ under fault-free conditions. If any of the statistics η_i are beyond the pre-determined thresholds, a gross error is detected.

3.5.3 Measurement Test (MT)

From Eqn. 3.26, one can define the vector of measurement adjustments:

$$\nu(k) = \mathbf{z}(k) - \hat{\tilde{\mathbf{z}}}(k) = \mathbf{R}_\sigma(\mathbf{M}^*)'\mathbf{R}_r^{-1}\mathbf{r}(k) \tag{3.30}$$

which follows a multivariate normal distribution under fault-free conditions.

The covariance matrix of $\nu(k)$ is:

$$\mathbf{R}_\nu = \mathbf{R}_\sigma(\mathbf{M}^*)'\mathbf{R}_r^{-1}\mathbf{M}^*\mathbf{R}_\sigma' \tag{3.31}$$

i.e. $\nu(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_\nu)$.

Therefore, the measurement test statistics can be defined as:

$$\eta_j = \frac{|\nu_j|}{\sqrt{\mathbf{R}_\nu(j,j)}} \tag{3.32}$$

which follows a standard normal distribution, i.e. $\eta_j \sim \mathcal{N}(0,1)$, under fault-free conditions. If any of the statistics η_j are beyond the pre-determined thresholds, a gross error is detected. Refer to [123] and [80] for detail.

3.5.4 Generalized Likelihood Ratio (GLR) Test

In contrast to other tests, the GLR test requires a process model in the presence of a gross error, i.e. the gross error model. The procedure is illustrated by Narasimhan and Mah (1987, [88]).

The gross error model for sensor faults can be represented as follows:

$$\mathbf{z}(k) = \tilde{\mathbf{z}}(k) + \sigma(k) + \mathbf{f}_z(k) \tag{3.33}$$

More generally, the gross error model for process faults such as leakage can be illustrated by modifying the model constraints:

$$\mathbf{M}^*\tilde{\mathbf{z}}(k) - \mathbf{M}^f\mathbf{f}(k) = \mathbf{0} \tag{3.34}$$

where $\mathbf{M}^f$ is a matrix representing the fault effects on the process correlation structure.

Using the above gross error models, it is possible to derive the statistical distribution of the constraint residuals when a gross error either in the measurements or constraints is present. It has been proved that under fault-free conditions, the residual follows a normal distribution with zero mean and covariance matrix $\mathbf{R}_r$. However, if a gross error is present, the residual will still follow a normal distribution with the same covariance matrix but with different expected value showing as below:

$$E\{\mathbf{r}(k)\} = \mathbf{M}^*\mathbf{f}_z(k) \quad \text{or} \quad E\{\mathbf{r}(k)\} = \mathbf{M}^f\mathbf{f}(k) \tag{3.35}$$

where $E\{\cdot\}$ denotes the expected value.

Therefore, one can formulate the following hypotheses test for gross error detection:

$$\begin{aligned} \mathcal{H}_0 &: \ \mu = \mathbf{0} \\ \mathcal{H}_1 &: \ \mu = \mathbf{M}^*\mathbf{f}_z(k) \text{ or } \mathbf{M}^f\mathbf{f}(k) \end{aligned} \tag{3.36}$$

where μ is the unknown expected value of $\mathbf{r}$.

In order to test these two hypotheses, one can use the likelihood ratio test. The likelihood ratio test statistic is given by

$$\lambda = \sup\frac{Pr\{\mathbf{r}|\mathcal{H}_1\}}{Pr\{\mathbf{r}|\mathcal{H}_0\}} \tag{3.37}$$

where $Pr\{\cdot|\cdot\}$ is the conditional probability.

Using the normal probability density function of $\mathbf{r}$, one can rewrite Eqn. 3.37 as follows:

$$\lambda = \sup\frac{\exp\{-0.5(\mathbf{r}-\mu)'\mathbf{R}_r^{-1}(\mathbf{r}-\mu)\}}{\exp\{-0.5\mathbf{r}'\mathbf{R}_r^{-1}\mathbf{r}\}} \tag{3.38}$$

Because λ is always positive, one can simplify Eqn. 3.38 by introducing another test statistic:

$$\eta = 2\ln\lambda = \sup\{\mathbf{r}'\mathbf{R}_r^{-1}\mathbf{r} - (\mathbf{r}-\mu)'\mathbf{R}_r^{-1}(\mathbf{r}-\mu)\} \tag{3.39}$$

Then, the test statistic η can be compared with a pre-determined threshold. A gross error is detected if η exceeds the threshold.

4

Fault Detectability Analysis for Linear Systems[1]

4.1 Introduction

Before designing a FDD scheme for a linear system, one must determine whether the detectability conditions are satisfied for the system in question, i.e. whether a fault signal can be detected when analytical redundancy exists in the system.

As indicated in Chapters 2 and 3, a fundamental element of a FDD system is the *residual generator*. The generated residual must be small (ideally zero) under fault-free conditions and large (or non-zero) if a fault is present. *Fault detectability* concerns the question of whether it is possible to construct a residual generator that

[1]The results in this chapter were presented at the *53rd Canadian Chemical Engineering Conference (CSChE)*, October 2003, Hamilton, Canada by Han and Shah.

is sensitive to any type of fault.

The importance of analyzing *fault detectability* has only recently been recognized by FDD community. For example, Saberi *et al.* ([115]) raised a number of fundamental questions in FDD systems in 1999. Nyberg and Nielson (2000, [92]) conducted a fault detectability analysis for both discrete and continuous time systems through the use of transfer functions.

4.2 Problem Definition

The objective of the research described in this chapter is to identify a condition under which - no matter what model-based fault detection method is used - a specific fault cannot be detected even when the process dynamics are exactly known. Therefore, the determinate property of a system is more relevant to the research topic. In contrast, the stochastic aspect of a system can only affect the sensitivity and the false alarm rate (FAR) of fault detection. No matter what method is used, the stochastic part of the system only affects the determination of the residual threshold, that is, the boundary between normal and abnormal regions.

Assume that the dynamics of a multivariate process can be represented by the following discrete time linear state space model with faults:

$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\tilde{\mathbf{u}}(k) + \mathbf{R}_1\mathbf{f}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{R}_2\mathbf{f}(k) \end{cases} \tag{4.1}$$

where $\{\mathbf{A},\ \mathbf{B},\ \mathbf{C},\ \mathbf{R_1},\ \mathbf{R_2}\}$ are the known true system matrices with appropriate dimensions.

In Eqn. 4.1, process noises, disturbances and other stochastic properties are not considered. Only the deterministic part of the process is taken into account. The reasons for formulating the problem in this manner are as follows:

1. This research is to determine how the system dynamics affect the fault detectability when the process dynamics are exactly known. Therefore, the model-plant mismatch (MPM) is not considered.

2. Residuals are generated for fault detection in most methods by using residual generators. If the residual is within some pre-determined threshold, no fault is present; otherwise, a fault is detected. If we eliminate the presence of noise and disturbance, the problem can be significantly simplified. This is so because,

in this case, non-zero residuals imply the presence of a fault. Thus, there is no need to discuss how to choose thresholds, which can be determined using various approaches.

3. Similar to the system properties of controllability and observability, fault detectability is a system property. This was pointed out by Nyberg and Nielson (1997, [93]). Detailed analysis of this feature will be presented later in this chapter.

4.3 Definitions

In order to characterize the fault detectability problem more precisely, pertinent definitions must be provided at outset. In this section, the definitions of "parity equation", "parity function" and "residual generator" are presented in preparation for further analysis. The terms "fault detectability" and "strong fault detectability" will then be defined mathematically.

4.3.1 Parity Equation and Parity Function

The terms "parity equations" and "parity function" are widely used within the field of "model-based fault detection". The definitions are provided by Chow and Willsky (1984, [19]). For the sake of completeness, those definitions are included below:

Definition 4.1 (Parity Equation) *A parity equation is an equation that can be written as*

$$\mathbf{M}(q)\tilde{\mathbf{y}}(k) + \mathbf{N}(q)\tilde{\mathbf{u}}(k) = \mathbf{0} \tag{4.2}$$

This equation is satisfied under the fault-free situation.

Definition 4.2 (Parity Function) *A parity function is a function that can be written as*

$$\mathcal{F}(k) = \mathbf{M}(q)\mathbf{y}(k) + \mathbf{N}(q)\tilde{\mathbf{u}}(k) \tag{4.3}$$

The value of this function is zero under fault-free situation.

where $\mathbf{M}(q)$ and $\mathbf{N}(q)$ denote matrices of polynomials in q. The left hand side of Eqn. 4.3 is also known as *parity vector*, which is non-zero if and only if a fault is present.

4.3.2 Residual Generator

In the field of fault detection, residual is a signal that is equal to zero when no fault occurs and non-zero when a fault occurs. The fault is generated by a system referred to as *Residual Generator*. Based on Definition 4.2, parity function has a close relationship with the residual generator.

Before introducing the detailed procedure, it is necessary to provide definition of *residual generator*, which will be used frequently in this chapter. The definition is given in the following:

Definition 4.3 (Residual generator) *A residual generator is a dynamic system that can be constructed in the following way:*

$$\mathbf{r}(k) = \mathbf{G}_y(q)\mathbf{y}(k) + \mathbf{G}_u(q)\tilde{\mathbf{u}}(k) \tag{4.4}$$

where the residual signal $\mathbf{r}(k)$ *is equal to null vector or* $\mathbf{0}$ *if* $\mathbf{f}(k) = \mathbf{0}$.

Based on Definition 4.3, the generated residuals can be used for fault detection. In order to formulate such a residual generator, some mathematical manipulations must first be performed.

After performing a series of recursions on Eqn.4.1, one arrives at

$$\mathbf{x}(k) = \mathbf{A}^s\mathbf{x}(k-s) + \sum_{i=k-s}^{k-1} \mathbf{A}^{k-1-i}\left[\mathbf{B}\tilde{\mathbf{u}}(i) + \mathbf{R}_1\mathbf{f}(i)\right] \tag{4.5}$$

and

$$\mathbf{y}(k) = \mathbf{C}\mathbf{A}^s\mathbf{x}(k-s) + \mathbf{C}\sum_{i=k-s}^{k-1} \mathbf{A}^{k-1-i}\left[\mathbf{B}\tilde{\mathbf{u}}(i) + \mathbf{R}_1\mathbf{f}(i)\right] + \mathbf{R}_2\mathbf{f}(k) \tag{4.6}$$

where s is the order of the *parity space* defined by Chow and Willsky (1984, [19]).

Further, by stacking, the following equation can be obtained:

$$\mathbf{y}_s(k) - \mathbf{H}_s\tilde{\mathbf{u}}_s(k) \;=\; \mathbf{\Gamma}_s\mathbf{x}(k-s) + \mathcal{H}_s\mathbf{f}_s(k) \tag{4.7}$$

where

$$\mathbf{\Gamma}_s = \left[\begin{array}{cccc} (\mathbf{C})' & (\mathbf{C}\mathbf{A})' & \cdots & (\mathbf{C}\mathbf{A}^s)' \end{array}\right]' \in \Re^{m_s \times n}$$

is the extended observability matrix; and

$$\mathcal{H}_s = \left[\begin{array}{cccc} \mathbf{R}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{s-1}\mathbf{R}_1 & \mathbf{C}\mathbf{A}^{s-2}\mathbf{R}_1 & \cdots & \mathbf{R}_2 \end{array}\right] \in \Re^{m_s \times p_s}$$

is a lower triangular block Toeplitz matrix with $m_s = m(s+1)$ and $p_s = p(s+1)$. $\mathbf{H}_s \in \Re^{m_s \times l_s}$ is the same as $\mathcal{H}_s$, except that $\mathbf{R}_1$ and $\mathbf{R}_2$ are replaced by $\mathbf{B}$ and $\mathbf{0}$ respectively with $l_s = l(s+1)$.

In addition, $\mathbf{y}_s(k) = \left[\begin{array}{ccc} \mathbf{y}(k-s)' & \cdots & \mathbf{y}(k)' \end{array} \right]' \in \Re^{m_s}$ is the stacked output vector; $\tilde{\mathbf{u}}_s(k) \in \Re^{l_s}$, $\mathbf{f}_s(k) \in \Re^{p_s}$ are similarly stacked as $\mathbf{y}_s(k)$.

One can select a transformation matrix $\mathbf{W}_o$, which is located in the *left null space* of $\mathbf{\Gamma}_s$, i.e. $\mathbf{W}_o\mathbf{\Gamma}_s \equiv \mathbf{0}$. Pre-multiplying both sides of Eqn. 4.7 by $\mathbf{W}_o$, the unknown state vector $\mathbf{x}(k-s)$ can be removed from the equation, leading to the following function:

$$\begin{aligned} \mathbf{r}(k) &\equiv \mathbf{W}_o\left(\mathbf{y}_s(k) - \mathbf{H}_s\tilde{\mathbf{u}}_s(k)\right) \\ &= \mathbf{W}_o\mathcal{H}_s\mathbf{f}_s(k) \end{aligned} \tag{4.8}$$

On the right hand side of Eqn. 4.8, the first line is the computational form and the second line is the internal form. From the internal form, it is easy to determine that $\mathbf{r}(k)$ is non-zero only if a fault exists. According to Definition 4.2, Eqn. 4.8 is a set of parity functions that depend on the different selection of $\mathbf{W}_o$ and s. Eqn. 4.8 has an attractive feature that shows up in the following theorem.

Theorem 4.1 *Eqn. 4.8 gives all the possible residual generators defined by Definition 4.3 for Eqn. 4.1 using the process inputs and outputs.*

Proof: Firstly, we need to prove Eqn. 4.8 is a residual generator defined by Definition 4.3.

From Eqn. 4.8, we can make the following manipulations:

$$\begin{aligned} \mathbf{r}(k) &= \mathbf{W}_o\left(\mathbf{y}_s(k) - \mathbf{H}_s\tilde{\mathbf{u}}_s(k)\right) \\ &= \mathbf{W}_o\begin{bmatrix} q^s\mathbf{y}(k) \\ q^{s-1}\mathbf{y}(k) \\ \vdots \\ q^0\mathbf{y}(k) \end{bmatrix} - \mathbf{W}_o\mathbf{H}_s\begin{bmatrix} q^s\tilde{\mathbf{u}}(k) \\ q^{s-1}\tilde{\mathbf{u}}(k) \\ \vdots \\ q^0\tilde{\mathbf{u}}(k) \end{bmatrix} \\ &= \left(\sum_{i=0}^{s} q^i\mathbf{W}_o(:,(s-i)m+1:(s-i)m+m)\right)\mathbf{y}(k) \\ &\quad - \left(\sum_{i=0}^{s} q^i\mathbf{W}_o\mathbf{H}_s(:,(s-i)m+1:(s-i)m+m)\right)\tilde{\mathbf{u}}(k) \end{aligned} \tag{4.9}$$

where $\mathbf{A}(:, m:n)$ represents the m^{th} to n^{th} columns of $\mathbf{A}$ matrix, which is consistent with the expression in *Matlab®*.

Let $\mathbf{G}_y(q) = \sum_{i=0}^{s} q^i \mathbf{W}_o(:, (s-i)m+1 : (s-i)m+m)$ and $\mathbf{G}_u(q) = \sum_{i=0}^{s} q^i \mathbf{H}_s(:, (s-i)m+1 : (s-i)m+m)$. Comparing Eqn. 4.9 with Definition 4.3, it is readily concluded that Eqn. 4.8 has the same structure defined for residual generator. Moreover, it is evident from the second line of Eqn. 4.8 that $\mathbf{r}(k)$ can be non-zero only if $\mathbf{f}_s(k) \neq \mathbf{0}$. Therefore, Eqn. 4.8 gives a residual generator.

The next step is to prove that any residual generator can be represented by Eqn. 4.8.

Based on the definition of residual generator, the residual should be non-zero if and only if a fault exists. Suppose we have a residual generator $\mathbf{r}(k) = \mathbf{G}_y(q)\mathbf{y}(k) + \mathbf{G}_u(q)\tilde{\mathbf{u}}(k)$, which is zero under fault-free conditions, i.e. $\mathbf{r}(k) = \mathbf{0}$, $\forall\, \mathbf{f}(k) = 0$. The reason for the existence of such a generator is due to analytical redundancy.

Under fault-free conditions, one can arrive at

$$\mathbf{r}(k) = \mathbf{G}_y(q)\mathbf{y}(k) + \mathbf{G}_u(q)\tilde{\mathbf{u}}(k) = \mathbf{D}_y^{-1}(q)\mathbf{N}_y(q)\mathbf{y}(k) + \mathbf{D}_u^{-1}(q)\mathbf{N}_u(q)\tilde{\mathbf{u}}(k) = \mathbf{0}$$

i.e.

$$\mathbf{D}_u(q)\mathbf{N}_y(q)\mathbf{y}(k) + \mathbf{D}_y(q)\mathbf{N}_u(q)\tilde{\mathbf{u}}(k) = \mathbf{0}$$

where $\mathbf{D}_y(q)$ and $\mathbf{D}_u(q)$, $\mathbf{N}_y(q)$ and $\mathbf{N}_u(q)$ are the denominators and numerators of transfer function matrices $\mathbf{G}_y(q)$ and $\mathbf{G}_u(q)$ respectively, and $\mathbf{D}_y(q)$ is assumed to be diagonal transfer function matrix.

Hence, the terms in front of $\mathbf{y}(k)$ and $\tilde{\mathbf{u}}(k)$ are finite-order polynomials. Then, s can be defined as the maximum order of these two polynomials. Therefore, the residual generator under fault-free conditions can be written as follows:

$$\left(\sum_{i=0}^{s} q^i \mathbf{A}_i\right)\mathbf{y}(k) + \left(\sum_{i=0}^{s} q^i \mathbf{B}_i\right)\tilde{\mathbf{u}}(k) = \mathbf{0}$$

where $\mathbf{A}_i$ and $\mathbf{B}_i$ are the coefficients of both polynomials.

We can also write the above equation in the stacking form:

$$\bar{\mathbf{A}}\mathbf{y}_s(k) + \bar{\mathbf{B}}\tilde{\mathbf{u}}_s(k) = \mathbf{0} \tag{4.10}$$

where $\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_s & \cdots & \mathbf{A}_1 & \mathbf{A}_0 \end{bmatrix}$ and $\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_s & \cdots & \mathbf{B}_1 & \mathbf{B}_0 \end{bmatrix}$.

From Eqn. 4.7, under fault-free conditions the above equation can be rewritten as

$$\bar{\mathbf{A}}\boldsymbol{\Gamma}_s\mathbf{x}(k-s) + \left(\bar{\mathbf{A}}\mathbf{H}_s + \bar{\mathbf{B}}\right)\tilde{\mathbf{u}}_s(k) = \mathbf{0}$$

In order for the above equation to hold true, the following two conditions must be satisfied simultaneously:

$$\bar{\mathbf{A}}\boldsymbol{\Gamma}_s = \mathbf{0}$$
$$\bar{\mathbf{A}}\mathbf{H}_s + \bar{\mathbf{B}} = \mathbf{0} \quad .$$

Since both $\bar{\mathbf{A}}$ and $\mathbf{W}_o$ are orthogonal to $\boldsymbol{\Gamma}_s$, we can let $\mathbf{W}_o = \bar{\mathbf{A}}$. Then, Eqn. 4.10 is exactly the same as Eqn. 4.8. This proves any residual generator can be represented in the format of parity space. ■

4.3.3 Fault Detectability and Strong Fault Detectability

Before discussing fault detectability criteria, the definition of fault detectability must be given explicitly. This problem has been identified by many researchers. Chen and Patton (1994, [15]) provided definitions of fault detectability and strong fault detectability as properties of a residual generator. The definitions are quoted as follows:

Definition 4.4 (Fault Detectability) *A fault f is detectable if the transfer function from the fault to the residual $G_{rf}(q)$ is non-zero:*

$$G_{rf}(q) \neq 0$$

Definition 4.5 (Strong Fault Detectability) *A fault f is strongly detectable if the transfer function from the fault to the residual $G_{rf}(q)$ satisfies:*

$$G_{rf}(1) \neq 0$$

Nyberg and Nielson (1997, [93]) pointed out a system's properties limit the possibilities for constructing a residual that is fault detectable and/or strongly fault detectable. Therefore, fault detectability must be regarded as a system property. Nyberg and Nielson gave their definitions of fault detectability and strong fault detectability as follows:

Definition 4.6 (Fault Detectability) *A fault is detectable in a system if and only if there exists a residual in which the fault is detectable according to Definition 4.4.*

Definition 4.7 (Strong Fault Detectability) *A fault is strongly detectable in a system if and only if there exists a residual in which the fault is strongly detectable according to Definition 4.5.*

It is readily concluded that all the above definitions focus on the characteristics of the transfer function between residual and fault. However, because it is difficult to assume the shape of fault signal before hand, the property of the transfer function alone is not enough to provide the overall solution to this problem. The reason is illustrated in the following example.

Example 4.1 Given a system

$$\begin{cases} \mathbf{x}(k+1) &= \begin{bmatrix} 2 & 1 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tilde{\mathbf{u}}(k) \\ \mathbf{y}(k) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} f^1(k) \\ f^2(k) \end{bmatrix} \end{cases}$$

where $\mathbf{f}(k) = \begin{bmatrix} f^1(k) \\ f^2(k) \end{bmatrix}$ represents the output sensor fault, which is the only fault considered in this example.

By selecting a parity space order that is equal to the process order, the matrices generated by stacking can be obtained as follows:

$$\mathbf{\Gamma}_2 = \begin{bmatrix} 1 & 0 & 2 & 0 & 4 & 0 \\ 0 & 1 & 1 & 0.5 & 2.5 & 0.25 \end{bmatrix}' \quad \textit{and} \quad \mathcal{H}_2 = \mathbf{I}_6$$

Thus, the residual generator obtained from Eqn. 4.8 can be seen in the following:

$$\mathbf{r}(k) = \mathbf{W}_o \mathbf{f}_2(k)$$

where $\mathbf{W}_o = \begin{bmatrix} -1 & 1 & 1 & 1 & -0.25 & -7.5 \end{bmatrix}$, which is orthogonal to $\mathbf{\Gamma}_2$.

Using the above definitions to check the detectability of this system, we can discover that this system should be not only detectable but also strongly detectable. However, up to now, there is no assumption about the shape of the fault signal. Therefore, we can make use of any type of fault. Thus, the following fault is assumed to be happening:

$$\begin{cases} f^1(k) &= \begin{cases} 0 & k < 0 \\ 2^k & k \geq 0 \end{cases} \\ f^2(k) &= 0 \qquad \forall\, k \end{cases}$$

Therefore, when $k \geq 2$, the following residual can be generated:

$$\begin{aligned} \mathbf{r}(k) &= -0.25f^1(k) + f^1(k-1) - f^1(k-2) \\ &= -0.25 \times 2^k + 2^{k-1} - 2^{k-2} = 0, \quad \forall\, k \geq 2 \end{aligned}$$

Obviously, this system is not strongly detectable. It is only detectable. This is so because the residual will stay at zero after the second time instant, even though the fault is still present.

Although the above definitions are not entirely accurate, they are important in research on fault detectability for the following reasons:

1. Different levels of detectability can be distinguished through the use of these definitions. These definitions identified the phenomenon of residual decay ([15]). From a practical perspective, residual decay should be avoided because it will not generate a continuous alarm signal even with the presence of a fault.

2. The detectability of a fault depends not only on the residual generator design, but also on the system's limitations. If a residual generator cannot detect a fault, it does not necessarily mean the fault is systematically undetectable. Lack of detectability may be due to poor design of the generator. Thus, from a system perspective, fault detectability should be defined as a property of the considered system rather than a residual generator.

In summary, Chen and Patton ([15]) pointed out the difference between fault detectability and strong fault detectability. Further, Nyberg and Nielson ([93]) defined (strong) fault detectability as a system property rather than the property due to the residual generator. However, both studies used the characteristic of the transfer function between the fault and the residual to make the decision. As illustrated in Example 4.1, nonzero static gain of $G_{rf}(q)$ cannot guarantee the final value of the residual to be nonzero for a fault signal other than a bias. This issue leads to the proposed definitions of fault detectability and strong fault detectability:

Definition 4.8 (Fault Detectability) *A fault is detectable in a system if and only if there exists a residual generator $\mathcal{R}$, whose output $\mathbf{r}(k)$ satisfies the following condition: if a fault occurs starting at k_0, i.e. $\mathbf{f}(k) \neq \mathbf{0}\ \forall\ k \geq k_0$, there exists a finite $k_1 \geq k_0$ where $\mathbf{r}(k_1) \neq \mathbf{0}$ regardless of the shape of the fault signal.*

Definition 4.9 (Strong Fault Detectability) *A fault is strongly detectable in a system if and only if there exists a residual generator $\mathcal{R}$, whose output $\mathbf{r}(k)$ satisfies the following condition: if a fault occurs starting at k_0, i.e. $\mathbf{f}(k) \neq \mathbf{0}\ \forall\ k \geq k_0$, there exists a finite $k_1 \geq k_0$ where $\mathbf{r}(k) \neq \mathbf{0}\ \forall\ k \geq k_1$ regardless of the shape of the fault signal.*

Definitions 4.8 and 4.9 define the properties of both levels of fault detectability by describing the behavior of the best available residual generator. If the residual turns to be non-zero after the presence of a fault, we can refer the fault as to be detectable. In contrast, if the residual turns and continues to be non-zero after the presence of a fault, then the fault is regarded as strongly detectable. Note that no assumption is made about the shape of fault signal. In the most cases, one cannot know the fault signal before it occurs. The purpose of (strong) fault detectability is to explore the following question: once one or more elements in the fault vector $\mathbf{f}(k)$ turn out to be non-zero, does the system structure limit the design of a residual generator that can guarantee detection of that fault? Based on Definitions 4.8 and 4.9, the following point must be made clear: if a fault is not (strongly) detectable, it does not mean one cannot design a residual generator that can (strongly) detect a particular type of fault; it only means one cannot design a residual generator that can (strongly) detect *any* type of fault.

4.4 Fault Detectability Analysis

Based on Theorem 4.1, it is clear that Eqn. 4.8 is equivalent to all the residual generators for a particular system. From this point onward in this chapter, Eqn. 4.8 is used directly as the residual generator. The criteria of fault detectability and strong fault detectability are investigated separately.

4.4.1 Condition for Fault Detectability

Note that $\mathbf{R}_1(:, i)$ and $\mathbf{R}_2(:, i)$ cannot be zero vectors simultaneously for any $1 \leq i \leq p$. If both $\mathbf{R}_1(:, i)$ and $\mathbf{R}_2(:, i)$ are zero vectors, then the i^{th} component of fault vector $\mathbf{f}(k)$ cannot affect the system dynamics. Hence, because of its complete decoupling from the system of interest, the fault is harmless.

The problem of fault detectability will now be investigated based on Definition 4.8. Several pertinent results can be obtained through the analysis. The condition for

fault detectability is shown in the following lemma and theorem.

Lemma 4.1 *The i^{th} fault is detectable if and only if*

$$\mathbf{W}_o\mathcal{H}_{s,i} \neq \mathbf{0} \tag{4.11}$$

$$\text{where } \mathcal{H}_{s,i} = \begin{bmatrix} \mathbf{R}_2(:,i) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{R}_1(:,i) & \mathbf{R}_2(:,i) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{s-1}\mathbf{R}_1(:,i) & \mathbf{C}\mathbf{A}^{s-2}\mathbf{R}_1(:,i) & \cdots & \mathbf{R}_2(:,i) \end{bmatrix} \in \Re^{m_s \times (s+1)}.$$

Proof: From Eqn.4.8, assume the i^{th} fault begins at time instant k_0, i.e. the remaining components of fault vector $\mathbf{f}(k)$ stay at zero. One can then arrive at

$$\mathbf{r}(k) = \mathbf{W}_o\mathcal{H}_{s,i}\mathbf{f}_s^i(k) \tag{4.12}$$

where $\mathbf{f}_s^i(k)$ is the stacked vector of i^{th} component of fault vector $\mathbf{f}(k)$.

If Eqn. 4.12 is rewritten in the polynomial format, the following relationship can be obtained by using the back shift operator q.

$$\mathbf{r}(k) = \left(\sum_{j=1}^{s+1} q^{s+1-j}\mathbf{W}_o\mathcal{H}_{s,i}(:,j)\right) f^i(k) \tag{4.13}$$

where $f^i(k)$ is the i^{th} component of $\mathbf{f}(k)$.

If Eqn. 4.11 is satisfied, there must exist a vector that $\mathbf{W}_o\mathcal{H}_{s,i}(:,\alpha) \neq \mathbf{0}$, $1 \leq \alpha \leq s+1$ and for any $\beta > \alpha$, $\mathbf{W}_o\mathcal{H}_{s,i}(:,\beta) = \mathbf{0}$, i.e., $\mathbf{W}_o\mathcal{H}_{s,i}(:,\alpha)$ is the last non-zero column of the matrix $\mathbf{W}_o\mathcal{H}_{s,i}$. Then, when $k = k_0 + s + 1 - \alpha$, the residual can be expressed as:

$$\begin{aligned} \mathbf{r}(k) &= \left(\sum_{j=1}^{\alpha} q^{s+1-j}\mathbf{W}_o\mathcal{H}_{s,i}(:,j)\right) f^i(k) = \mathbf{W}_o\mathcal{H}_{s,i}(:,\alpha)q^{s+1-\alpha}f^i(k) \\ &= \mathbf{W}_o\mathcal{H}_{s,i}(:,\alpha)f^i(k_0) \neq \mathbf{0} \end{aligned}$$

Therefore, the i^{th} fault is detectable and $s+1-\alpha$ is the time delay for this residual generator to detect the fault.

Moreover, if the i^{th} fault is detectable, i.e. $\exists k_1 \geq k_0$, $\mathbf{r}(k_1) \neq \mathbf{0}$, the matrix $\mathbf{W}_o\mathcal{H}_{s,i}$ cannot be zero. ∎

Two important questions become apparent as a consequence of the above: can one be certain that the matrix $\mathbf{W}_o\mathcal{H}_{s,i}$ never becomes zero under some condition? If so, is that condition related to the residual generator design or to the process limitation? The following theorem is given and proved to resolve these questions.

Theorem 4.2 *Any fault described in Eqn. 4.1 is detectable.*

Proof:

Assume i^{th} fault begins at time instant k_0. From Lemma 4.1, the detectability of i^{th} fault is equivalent to determining whether the condition $\mathbf{W}_o\mathcal{H}_{s,i} \neq \mathbf{0}$ is satisfied. It is already known that $\mathcal{H}_{s,i}$ cannot be a null matrix, this detectability condition can be rewritten in another way: $\mathbf{W}_o \not\subset \mathcal{H}_{s,i}^{\perp}$.

Furthermore, the matrix $\mathbf{W}_o$ is designed to be orthogonal to the extended observability matrix $\mathbf{\Gamma}_s$, i.e. $\mathbf{W}_o \subset \mathbf{\Gamma}_s^{\perp}$. Therefore, it can be concluded that if and only if $\mathcal{H}_{s,i}$ is not within the range space of $\mathbf{\Gamma}_s$, i.e. $\mathcal{H}_{s,i} \not\subset \mathcal{R}(\mathbf{\Gamma}_s)$ where $\mathcal{R}(\cdot)$ denotes the range space of a matrix, the corresponding fault then can be detected.

The matrix $\mathcal{H}_{s,i}$ has the dimension of m_s by $s+1$. However, the rank of this matrix varies under different situations. The following result can be obtained:

$$rank\,(\mathcal{H}_{s,i}) = s + 1 - d_i$$

where d_i is defined as

$$d_i = \begin{cases} 0 & if\ \mathbf{R}_2(:,i) \neq \mathbf{0} \\ \alpha + 1 & if\ \mathbf{C}\mathbf{A}^{\alpha}\mathbf{R}_1(:,i) \neq \mathbf{0}\ and\ \mathbf{C}\mathbf{A}^{\beta}\mathbf{R}_1(:,i) = \mathbf{0},\ \forall\, \beta < \alpha\ \&\ \alpha \geq 0 \end{cases}$$

Physically, d_i is the time delay from the i^{th} fault to the output.

Because it is assumed that the system itself is observable, the rank of matrix $\mathbf{\Gamma}_s$ is $rank\,(\mathbf{\Gamma}_s) = n$ given that $s \geq n$. Therefore, by selecting proper parity space order s, one can design a residual generator such that $rank\,(\mathcal{H}_{s,i}) > rank\,(\mathbf{\Gamma}_s)$, which guarantees $\mathcal{H}_{s,i} \not\subset \mathcal{R}(\mathbf{\Gamma}_s)$. Thus, it is proved that any fault can be detected if the right residual generator is used. ■

For people working in the FDD area, it is exciting to know that any fault can be detected systematically. However, as pointed out by many researchers and as is evident in Definitions 4.8 and 4.9, knowing that a fault is detectable is not sufficient in real-world situations. The generated residual could decay to zero even though the fault is still present. In order to avoid the problem of residual decay, it is important to ensure a fault is strongly detectable. The condition for strong fault detectability will be discussed in the next section.

4.4.2 Condition for Strong Fault Detectability

In this section, the condition for strong fault detectability is discussed. This issue has been discussed by many researchers. Before analyzing strong fault detectability,

several manipulations must be done. For the sake of simplicity, define $\bar{\mathcal{H}}_{s,i}$ as the matrix consisting of all non-zero columns of $\mathcal{H}_{s,i}$, i.e. $\bar{\mathcal{H}}_{s,i} \in \Re^{m_s \times (s+1-d_i)}$, and $\bar{\mathbf{f}}_s^i(k)$ is the corresponding fault vector. Based on the proof of Theorem 4.2, the matrix $\bar{\mathcal{H}}_{s,i}$ has the following feature:

Lemma 4.2 *Rank* $\left(\bar{\mathcal{H}}_{s,i}\right) = s + 1 - d_i$, *where* d_i *is the time delay from the* i^{th} *fault to the process outputs.*

Proof: See the proof of Theorem 4.2. ∎

The following theorem is given as the condition for strong fault detectability:

Theorem 4.3 *Fault* $f^i(k)$ *is strongly detectable if and only if at least one column in* $\bar{\mathcal{H}}_{s,i}$ *cannot be expressed as a linear combination of remaining columns in matrix* $\left[\begin{array}{cc} \bar{\mathcal{H}}_{s,i} & \mathbf{\Gamma}_s \end{array}\right]$.

Proof: The proof starts with the *only-if* aspect.

Assume that the fault $f^i(k)$ is strongly detectable. Based on Definition 4.9, a finite $k_1 > k_0$ must exist so that $\mathbf{r}(k) \neq \mathbf{0}$, $\forall\ k \geq k_1$. Without loss of generality, one can assume $k_1 > k_0 + s$.

Note the special property of vector $\mathbf{f}_s^i(k)$ when $k > k_0 + s$: each element in vector $\bar{\mathbf{f}}_s^i(k)$ is non-zero. This differs from non-zero vectors and in this chapter this special vector is referred as to *strict non-zero vector*.

Because fault $f^i(k)$ is strongly detectable, the residual $\mathbf{r}(k) = \mathbf{W}_o \bar{\mathcal{H}}_{s,i} \bar{\mathbf{f}}_s^i(k) \neq \mathbf{0}$, $\forall\ k > k_0 + s$. This can be rewritten as:

$$\nexists \text{ strict nonzero vector } \bar{\mathbf{f}}_s^i(k) \text{ such that } \bar{\mathcal{H}}_{s,i} \bar{\mathbf{f}}_s^i(k) \not\subseteq \mathcal{R}(\mathbf{\Gamma}_s)$$

If the vector $\bar{\mathbf{f}}_s^i(k)$ can be any ordinary non-zero vectors, this problem has a straightforward solution: $\left[\begin{array}{cc} \bar{\mathcal{H}}_{s,i} & \mathbf{\Gamma}_s \end{array}\right]$ is of full column rank. However, for strict non-zero vectors, we may obtain a more flexible condition. That is to say, $\left[\begin{array}{cc} \bar{\mathcal{H}}_{s,i} & \mathbf{\Gamma}_s \end{array}\right]$ of full column rank is only a sufficient condition for $f^i(k)$ being strongly detectable.

Assume $\left[\begin{array}{cc} \bar{\mathcal{H}}_{s,i} & \mathbf{\Gamma}_s \end{array}\right]$ is not of full column rank. One can then find a non-empty right null space that satisfies the following:

$$\left[\begin{array}{ccc} \bar{\mathcal{H}}_{s,i}^{(1)} & \bar{\mathcal{H}}_{s,i}^{(2)} & \mathbf{\Gamma}_s \end{array}\right] \left[\begin{array}{c} \mathbf{I} \\ \mathbf{M} \\ \mathbf{N} \end{array}\right] = \mathbf{0} \tag{4.14}$$

i.e.

$$\begin{bmatrix} \bar{\mathcal{H}}_{s,i}^{(1)} & \bar{\mathcal{H}}_{s,i}^{(2)} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{M} \end{bmatrix} = -\mathbf{\Gamma}_s \mathbf{N} \tag{4.15}$$

where $\begin{bmatrix} \bar{\mathcal{H}}_{s,i}^{(2)} & \mathbf{\Gamma}_s \end{bmatrix}$ is of full column rank, $\bar{\mathcal{H}}_{s,i}^{(1)}$ is the remaining columns in $\bar{\mathcal{H}}_{s,i}$ and $\begin{bmatrix} \mathbf{I} & \mathbf{M}' & \mathbf{N}' \end{bmatrix}'$ is the right null space of $\begin{bmatrix} \bar{\mathcal{H}}_{s,i} & \mathbf{\Gamma}_s \end{bmatrix}$.

Therefore, the original problem can be expressed as: $\forall\, \mathbf{v} \neq \mathbf{0}$, (1) $\begin{bmatrix} \mathbf{I} & \mathbf{M}' \end{bmatrix}' \mathbf{v}$ is not a strict non-zero vector, i.e. at least one element of the vector is zero, and (2) $\mathbf{N}\mathbf{v} \neq \mathbf{0}$. The first condition can be satisfied only when $\mathbf{M}\mathbf{v}$ has at least one zero element for all strict non-zero vectors $\mathbf{v}$. Hence, it concludes that $\mathbf{M}$ has at least one zero row. The second condition can be easily proved if $\bar{\mathcal{H}}_{s,i}$ is of full column rank.

By rearranging the rows of matrix $\mathbf{M}$, it can be written as $\tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M}_1' & \mathbf{0} \end{bmatrix}'$. Then matrix $\bar{\mathcal{H}}_{s,i}^{(2)}$ can be formulated as $\bar{\mathcal{H}}_{s,i}^{(2)} = \begin{bmatrix} \tilde{\mathcal{H}}_{s,i}^{(2,1)} & \tilde{\mathcal{H}}_{s,i}^{(2,2)} \end{bmatrix}$. From Eqn. 4.14, it is readily determine that $\bar{\mathcal{H}}_{s,i}^{(1)} = \tilde{\mathcal{H}}_{s,i}^{(2,1)} \mathbf{M}_1 + \mathbf{\Gamma}_s \mathbf{N}$, i.e. $\bar{\mathcal{H}}_{s,i}^{(1)}$ can be expressed as a linear combination of $\begin{bmatrix} \tilde{\mathcal{H}}_{s,i}^{(2,1)} & \mathbf{\Gamma}_s \end{bmatrix}$. Because $\begin{bmatrix} \bar{\mathcal{H}}_{s,i}^{(2)} & \mathbf{\Gamma}_s \end{bmatrix}$ is of full column rank, $\tilde{\mathcal{H}}_{s,i}^{(2,2)}$ cannot be expressed as a linear combination of remaining columns of $\begin{bmatrix} \bar{\mathcal{H}}_{s,i} & \mathbf{\Gamma}_s \end{bmatrix}$.

The *if* aspect of the proof is as follows:

Assume at least one column of $\bar{\mathcal{H}}_{s,i}$ cannot be expressed as a linear combination of remaining columns of $\begin{bmatrix} \bar{\mathcal{H}}_{s,i} & \mathbf{\Gamma}_s \end{bmatrix}$. Thus, we can refer that column as $\mathbf{h}$ and the remaining columns in $\bar{\mathcal{H}}_{s,i}$ as $\bar{\mathbf{h}}$.

Therefore, it does not exist a vector $\begin{bmatrix} \mathbf{v}_1' & \mathbf{v}_2' \end{bmatrix}'$ such that $\mathbf{h} = \begin{bmatrix} \bar{\mathbf{h}} & \mathbf{\Gamma}_s \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$.

That is, it does not exist vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ such that $\begin{bmatrix} \mathbf{h} & \bar{\mathbf{h}} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{v}_1 \end{bmatrix} = \mathbf{\Gamma}_s \mathbf{v}_2$. Thus, for any strict non-zero vector $\mathbf{f}_{s,k}^i$, $\mathcal{H}_{s,i} \mathbf{f}_{s,k}^i \not\subseteq \mathcal{R}(\mathbf{\Gamma}_s)$, which can guarantee that the fault f_k^i is strongly detectable. ∎

4.4.3 Strong Fault Detectability for Output Sensor Faults

Theorem 4.3 provides the general solution for the strong detectability criterion. However, if only certain special types of fault are considered – that is, if special matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ are used – one can arrive at simpler conditions. Therefore, we have the following corollaries for output sensor faults.

Corollary 4.1 *The i^{th} output sensor fault is strongly detectable if and only if the system $\{\mathbf{A}, \bar{\mathbf{C}}_i\}$ is observable, where $\bar{\mathbf{C}}_i = \begin{bmatrix} \mathbf{C}(1:i-1,:) \\ \mathbf{C}(i+1:m,:) \end{bmatrix}$, i.e. the matrix $\mathbf{C}$ without i^{th} row.*

Proof: Because only output sensor faults are considered, $\mathbf{R}_1 = \mathbf{0}$ and $\mathbf{R}_2 = \mathbf{I}_m$. Therefore, the matrix $\mathcal{H}_{s,i}$ can be denoted as

$$\mathcal{H}_{s,i} = \begin{bmatrix} \mathbf{e}_i & & \\ & \ddots & \\ & & \mathbf{e}_i \end{bmatrix}$$

where $\mathbf{e}_i$ is the i^{th} column of identity matrix $\mathbf{I}$.

Thus, based on Theorem 4.3, to enable the i^{th} output sensor fault to be strongly detectable, at least one column in $\mathcal{H}_{s,i}$ cannot be expressed as a linear combination of remaining columns in $\begin{bmatrix} \mathcal{H}_{s,i} & \mathbf{\Gamma}_s \end{bmatrix}$. Denote this column as the j^{th} column, i.e. $\mathbf{h} = [\mathbf{0} \ \cdots \ \mathbf{e}_i' \ \cdots \ \mathbf{0}]'$, and the remaining columns as $\bar{\mathbf{h}}$. Hence, the condition of strong detectability is:

$$\nexists\, \mathbf{v} \neq \mathbf{0}, \text{ such that } \mathbf{h} = [\bar{\mathbf{h}} \ \ \mathbf{\Gamma}_s]\mathbf{v} \tag{4.16}$$

where $\mathbf{v} = [\mathbf{v}_1' \ \ \mathbf{v}_2']'$, $\mathbf{v}_1 = [\alpha_1 \ \cdots \ \alpha_s]' \in \Re^s$ and $\mathbf{v}_2 \in \Re^n$.

That is, $\nexists\, \mathbf{v} \neq \mathbf{0}$, in which case the following equations are satisfied simultaneously:

$$\begin{aligned} \mathbf{0} &= \alpha_1\mathbf{e}_i + \mathbf{C}\mathbf{v}_2 \\ \mathbf{0} &= \alpha_2\mathbf{e}_i + \mathbf{C}\mathbf{A}\mathbf{v}_2 \\ &\vdots \\ \mathbf{e}_i &= \mathbf{C}\mathbf{A}^{j-1}\mathbf{v}_2 \\ &\vdots \\ \mathbf{0} &= \alpha_s\mathbf{e}_i + \mathbf{C}\mathbf{A}^s\mathbf{v}_2 \end{aligned} \tag{4.17}$$

i.e.

$$\begin{bmatrix} -\alpha_1\mathbf{e}_i \\ \vdots \\ \mathbf{e}_i \\ \vdots \\ -\alpha_s\mathbf{e}_i \end{bmatrix} = \mathbf{\Gamma}_s\mathbf{v}_2 \tag{4.18}$$

The left hand side of Eqn. 4.18 is a vector with at least ms zero elements. In addition, note that α_i, $\forall\ i = [1, s]$, can have any value. Therefore, the issue can be rewritten as:

$\exists\ \mathbf{v}_2 \neq \mathbf{0}$, such that the following equations are satisfied simultaneously:

$$\bar{\mathbf{\Gamma}}_{s,i}\mathbf{v}_2 = \mathbf{0} \tag{4.19}$$

$$\mathbf{C}(i,:)\mathbf{A}^{j-1}\mathbf{v}_2 \neq 0 \tag{4.20}$$

where $\bar{\mathbf{\Gamma}}_{s,i} = [\bar{\mathbf{C}}_i' \ \cdots \ (\bar{\mathbf{C}}_i\mathbf{A}^s)']'$.

Based on the assumption that the original system is observable, i.e. $rank(\mathbf{\Gamma}_s) = n$, if Eqn. 4.19 is satisfied, then there must exist a j that satisfies Eqn. 4.20. Hence, to make the i^{th} output sensor fault strongly detectable, a non-zero vector $\mathbf{v}_2$ must not exist such that Eqn. 4.19 can be satisfied. One then can conclude that the matrix $\bar{\mathbf{\Gamma}}_{s,i}$ must be of full column rank. That is, $\{\mathbf{A}, \bar{\mathbf{C}}_i\}$ must be observable. ■

Corollary 4.1 states that if one wants to confirm the strong detectability of an output sensor fault, doing so is equivalent to confirming the remaining system's observability. This conclusion is consistent with what one would intuit. If the remaining system is not observable, i.e. some states of the system cannot be observed by the fault-free output sensors, one cannot infer the right measurement of the i^{th} output. Thus, it is impossible to determine whether the measurement of the i^{th} output sensor is faulty.

4.5 Numerical example

In the study by Chen and Patton (1994, [15]), an inverted pendulum example was used to show the difference between detectability and strongly detectability. This example is reinvestigated in this chapter by means of Theorems 4.2 and 4.3.

Example 4.2

The system described by Chen and Patton (1994, [15]) was in a continuous time domain. The discrete time realization of the system with sampling rate $T_s = 0.01sec$ can be represented as:

$$\mathbf{A} = \begin{bmatrix} 1 & -9.584 \times 10^{-5} & 0.009901 & 1.269 \times 10^{-7} \\ 0 & 1.002 & 0.0003109 & 0.009998 \\ 0 & -0.01911 & 0.9803 & -6.749 \times 10^{-6} \\ 0 & 0.3683 & 0.06197 & 1.0001 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -1.592 \times 10^{-5} \\ -5.077 \times 10^{-5} \\ -0.003174 \\ -0.01019 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

In this example, only output sensor faults are considered. Therefore, $\mathbf{R}_1 = \mathbf{0}$ and $\mathbf{R}_2 = \mathbf{I}_3$. Because from Theorem 4.2 all the fault is detectable, we are more focused on the strong fault detectability criterion.

By examining the matrix $\begin{bmatrix} \mathcal{H}_{4,i} \\ \mathbf{\Gamma}_4 \end{bmatrix}$, $\forall\, i = [1,3]$, with respect to Theorem 4.3, the first output sensor fault is not strongly detectable but the other two output sensor faults are.

Because in this example only an output sensor fault is considered, one can also use Corollary 4.1 to determine strong detectability. When $i = 1$, the remaining system cannot observe the first state. That is, the first output sensor fault is not strongly detectable. However, for $i = 2, 3$, the output sensor faults are strongly detectable.

4.6 Conclusion

The fundamental problem of fault detectability was investigated in this chapter. This problem has been categorized into two levels: detectability and strong detectability. The definitions are explicitly given. Conditions regarding fault detectability and strong fault detectability are also given and proved. The condition of strong fault detectability for output sensors is also derived as a special case of the general condition of strong fault detectability. Further, the theorems obtained are applied to a well-known inverted pendulum example.

5

Process Modelling for Pulp Bleaching Process[2]

5.1 Introduction

In pulp bleaching processes, most of the final product qualities for example, brightness are measured in the laboratory. Laboratory analysis introduces time delays and non-uniform samplings. Due to lack of real-time measurements of product qualities, the entire bleaching process is usually under manual control. As a result, it is often difficult to provide reliable, fast, online measurements essential to control product quality, which in turn can lead to excessive deviation from acceptable standards,

[2]A version of this chapter is currently under review by the *Nordic Pulp & Paper Research Journal* with the title of "Softsensor development for a thermo-mechanical pulp mill using Partial Least Squares" authored by Han, Shah, Pakpahan, Patwardhan and Robson.

especially when changing from one operating point to another. Therefore, a scheme to estimate these quality variables at regular sampling rates, in-between laboratory samples, is necessary for satisfactory quality control and process monitoring. The concept of *softsensor* (also known as inferential estimator) is proposed in this chapter to develop such a scheme without introducing expensive instrumentation. This scheme employs a mathematical model developed from the process of interest and historical process data to obtain realtime estimates of product quality variables.

The physio-chemical mechanics of pulp bleaching processes within the pulp and paper industry are poorly understood. Consequently, the bleaching process is often treated as a black–box. Considering the complexity of bleaching processes, the first principle modelling approach is not a trivial exercise. Alternatively, a data-based empirical modelling approach can be used to develop the process model. Multivariate statistical regression methods, such as Partial Least Squares (PLS, [35]), Canonical Variate Analysis (CVA, [52] & [53]) and Principal Component Regression (PCR, [87]), are frequently used for this purpose. Traditional identification methods, such as Prediction Error Method (PEM, [72]), Instrumental Variable Method (IVM, [121]) and Subspace Identification Method (SIM, [98]), are also widely used for dynamic modelling.

Identifying an industrial process by directly applying the above-mentioned technologies is not trivial for several reasons. The process under consideration is a time-delay dominant process with little dynamics and significant input nonlinearities. In addition, many process variables are measured for control or monitoring purposes. Nevertheless, due to the complexity of the process, one must determine the "optimal" set of variables to be used as model inputs. Identifying an industrial process is a difficult task involving various challenges. Those challenges are not addressed systematically in the literature.

This chapter outlines a procedure for building steady state softsensor models of quality variables such as brightness. The procedure employs the PLS method based on process data obtained from Millar Western Forest Products Ltd., Whitecourt, Alberta, Canada. The plant applies a Bleached Chemi-ThermoMechanical Pulp (BCTMP) process. The data used in this chapter are normal operating data readily available from plant historian databases. Process knowledge and correlation analysis are used to determine appropriate time delays for each process variable. Several nonlinear transformations are considered during the softsensor development stage to ensure that the overall model reflects the true relationship of the process. Further, the stepwise regression method is used to identify variables significantly

impacting important quality specifications, and the results are consistent with process knowledge. High correlation coefficients between predicted qualities and real measurements have been achieved in this chapter.

5.2 Concept of Softsensor

Softsensor is a powerful and increasingly-applied methodology that enables one to infer important process quality variables from other readily-accessed process variables such as pressure, flow rate and temperature. In industry, several measurements are used to monitor and ultimately control and optimize processes in order to achieve product consistency. Automatic control and optimization require regular and reliable measurements at the appropriate sampling frequency. Difficulties in measuring quality variables inevitably result in poor (or even no) control. Measurement difficulties can be due to various reasons:

- *Lack of appropriate online instruments.* Therefore, normal process operation must depend on laboratory tests, which can result in irregular data and long analysis time delays. Depending on how the laboratory analysis is conducted, the data may also be unreliable.

- *Reliability of online instruments.* Online sensors may be available but may suffer from long measurement delays (e.g. gas chromatographs) or may be subject to factors that affect the reliability of the sensor (e.g. drifts and fouling).

In either case, automatic control and optimization schemes cannot be implemented and process performance may be degraded as a result.

A softsensor is a mathematical model that describes the relationship between two blocks of variables: process variables and quality variables. After model has been developed, one can use measured process variables (softsensor inputs) to predict product quality variables (softsensor outputs). The softsensor design technique provides an easy and economical way to estimate quality variables as frequently as process variables are measured. The softsensor estimates can then be used for control and monitoring purposes. Softsensors can be developed based on various criteria. For example, there are dynamic vs. steady state softsensors, linear vs. nonlinear softsensors and so forth. One can decide what type of softsensors to develop based on the process characteristics and project requirements.

Softsensor development differs from the traditional system identification for control purposes. The ultimate objective of a softsensor model is to estimate quality variables accurately. When building such a model, one must include as many softsensor inputs as possible to consider a maximum range of effects on quality variables. Nevertheless, the traditional system identification for control purposes usually only models the effects from the manipulated variables (MVs) and major disturbances to the controlled variables (CVs).

The softsensor technique offers a number of advantages:

- *Easy to develop.* There are various modelling algorithms available for developing softsensor models. One can choose the appropriate algorithm based on process characteristics and project requirements.
- *Easy to implement.* To implement a softsensor model online, only a computer, which can read and write realtime data, is required. No other hardware and/or software is required.
- *Easy to maintain.* To maintain a softsensor, one only needs to apply the algorithm by using the latest data, determine what changes are required, and implement the new or revised model in the computer.
- *Good estimation results.* As will be shown in this chapter, the resulting estimates are highly consistent with laboratory measurements.
- *Economical.* Design and implementation of a softsensor model to provide realtime measurement of quality variables are usually more economical than installing complex hardware with similar performance.

5.3 Systematic Approach for Softsensor Design

In this section, a systematic approach to developing a softsensor is outlined. During the design procedure, understanding the process and having design experience are very important. These enable one to find a satisfactory solution more quickly and efficiently.

The softsensor development procedure presented here is standardized and suitable for application to any bleaching process. This procedure is summarized in Figure 5.1. Note that the procedure is iterative: one usually must go through the procedure several times to arrive at a refined model that satisfies the requirements.

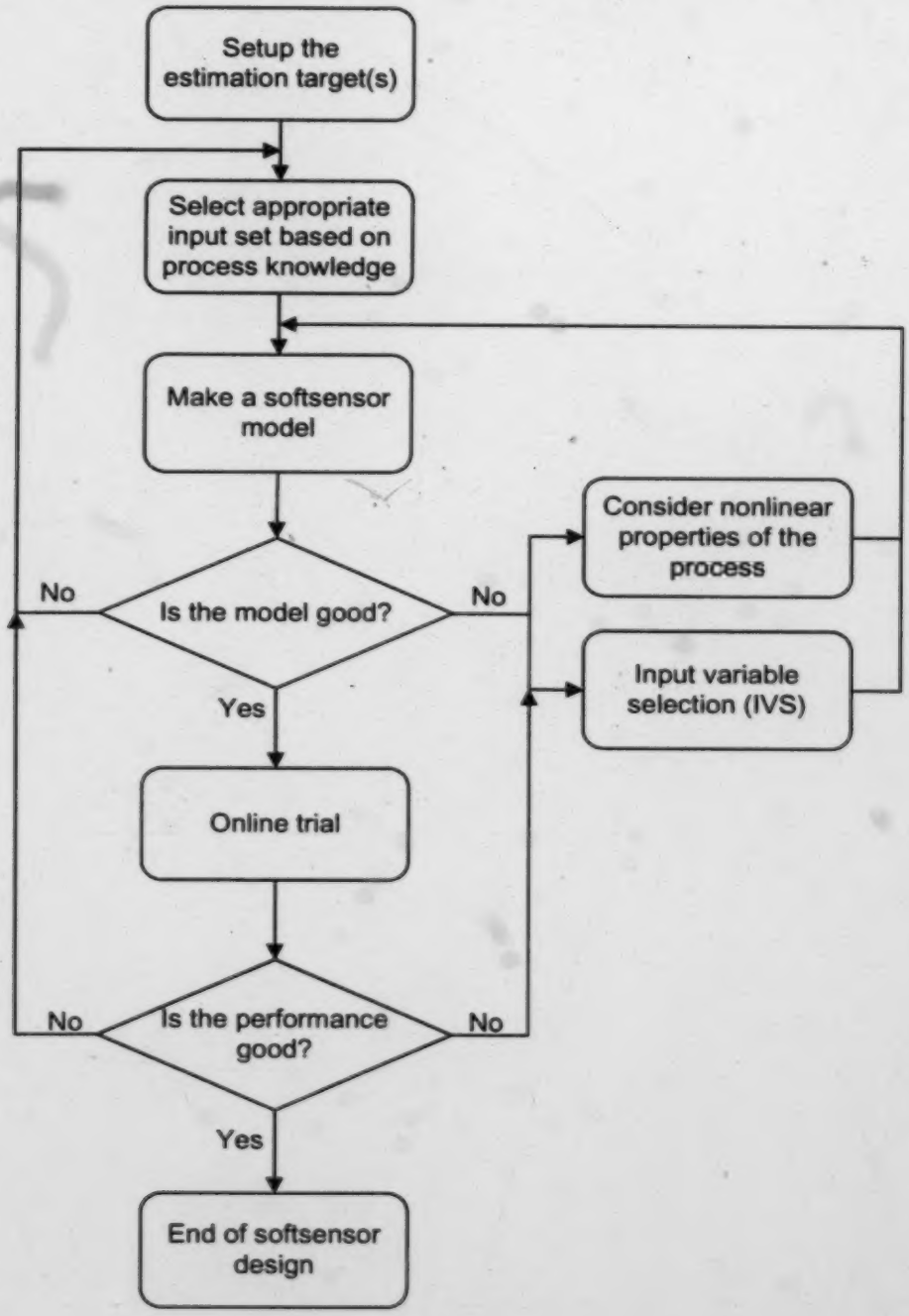

Figure 5.1: Softsensor development procedure

5.3.1 Modelling Method Selection

To design a softsensor for a given process, one must first decide which modelling method is most suitable for a given project. To do so, one must begin by classifying the available methods into categories. Then, based on the projects requirements and the process characteristics, one can determine which method is most suitable:

- *Linear vs. nonlinear approaches:* Linear modelling methods are suitable for close-to-linear processes or nonlinear processes that can be transformed into a linear framework with a known transformation. Due to their simplicity, linear approaches are widely applied in various applications ([73] [26]). On the other hand, if the process is highly nonlinear and the nonlinearity is totally unknown, one must try nonlinear approaches such as neural networks ([48]). Applications through nonlinear identification approaches are also available ([13] [1]). However, the implementation of the developed nonlinear models may require additional software, which increases costs.

- *Steady state vs. dynamic approaches:* The choice depends on whether the process has a large time constant compared to the input sampling rate. If the process is fast, steady state modelling may be enough. Otherwise, dynamic modelling is likely to perform better as it allows one to introduce more parameters into the modelling procedure.

- *Closed loop vs. open loop approaches:* Under some conditions, closed loop approaches enable one to develop an open loop model by using closed loop data. The closed loop system must be excited by either the setpoint change or process disturbances. On the other hand, to apply open loop approaches, one must break all the control loops to run the experiment, and that may cause serious operation problems and off-specification products. Thus, the approach chosen depends on whether the plant is available for experiment. Usually, due to the specially designed experiment, the open loop approach provides a better model.

5.3.2 Quality Indicators

After a model has been developed, one must evaluate its effectiveness. In order to do so, evaluation benchmarks must be established first. In this chapter, two benchmarks are used to evaluate the model: Cross-correlation Coefficient (CC) and Mean Squared

Error (MSE). Both are based on the estimated output ($\hat{\mathbf{Y}}$) and the measured output ($\mathbf{Y}$).

The cross-correlation coefficient can be derived by using the following equation:

$$CC(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{cov(\mathbf{Y}, \hat{\mathbf{Y}})}{\sqrt{var(\mathbf{Y})var(\hat{\mathbf{Y}})}} \tag{5.1}$$

where $cov(\mathbf{a}, \mathbf{b})$ denotes the cross-covariance between vectors $\mathbf{a}$ and $\mathbf{b}$, and $var(\mathbf{a})$ gives the variance of vector $\mathbf{a}$. The CC value is between -1 and 1. The higher the CC value, the better the prediction will match the measurement. The advantage of using CC as a benchmark is that the CC value has been scaled into the range of -1 and 1, thereby enabling us to judge the model's quality even without comparison to others. However, CC only represents the linear dependency between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. A high CC value does not mean the estimation error is small; it only indicates that the two variables are linearly dependent.

The MSE benchmark can be calculated as follows:

$$MSE(\mathbf{Y}, \hat{\mathbf{Y}}) = \sqrt{\frac{\sum_{i=1}^{N} (\mathbf{Y}(i) - \hat{\mathbf{Y}}(i))^2}{N}} \tag{5.2}$$

where N is the length of the data set. MSE represents the average absolute value of the estimation error as a function of time. Therefore, the smaller the MSE, the better the model. In contrast to CC, the MSE value is not scaled; consequently, it is difficult to determine whether a model is effective based only on its MSE value. However, from the MSE value, one can assess the accuracy of model estimation directly rather than the linear dependency as CC benchmark.

As mentioned above, MSE and CC indicate different types of model performance, and have their own advantages and disadvantages. Therefore, one must use both benchmarks to obtain the most comprehensive evaluation possible.

5.3.3 Input Variable Selection (IVS)

To design softsensor models for a special industrial process, usually all the measured process variables are provided as inputs at the start of the project. However, the inclusion of more process variables does not necessarily result in better performance of the softsensor models. This is the case because not all variables are relevant to the estimation target, i.e. softsensor outputs. Instead, it is highly possible that when the

irrelevant variables are removed, the model's performance will improve. If irrelevant variables are kept in the model, the perturbation and noise within these variables can potentially reduce the estimation accuracy very significantly. Therefore, it is imperative to select only those process variables that are relevant to the estimation targets. Process knowledge is critical to the elimination of irrelevant variables. However, further investigation is required to determine the relevant variables.

The problem of variable selection has been investigated in regression analysis by Hocking (1976, [50]). The two most common methods of variable selection are:

- *Forward selection.* One starts the model building process with a small set of input variables, and adds more variables if they improve predictability.
- *Backward selection.* One starts building the model with all possible input variables included, and removes those that do not improve predictability.

Theoretically, the more input variables included in a model, the larger the prediction variance and the lower the prediction bias. Because in this project, we are most interested in accurate prediction that is lower prediction bias, a backward selection method – stepwise regression - is used to determine the optimal input set.

The procedure of stepwise regression can be explained by the following steps:

1. Existing Model A with l inputs and 1 output
2. From model A, remove the 1^{st} input and build a new model based on the remaining inputs and the output
3. From model A, remove the 2^{nd} input and build a new model based on the remaining inputs and the output
4. .....
5. From model A, remove the l^{th} input and build a new model based on the remaining inputs (i.e. the 1^{st} to the $(l-1)^{th}$ inputs) and the output
6. Compare all the models obtained, determine which one is best based on either CC or MSE, and discard the corresponding input
7. Determine whether the best model has a satisfactory performance; if the answer is yes, stop; otherwise, continue
8. Use the best model as the model A, and back to step (2)

The stepwise regression method is iterative. After each iteration, the input set is reduced by one, resulting in a new model and its CC or MSE value. In order to further illustrate the idea of stepwise regression, a block diagram is shown in Figure 5.2.

The stepwise regression method gives a sub-optimal solution of the following combinatorial optimization problem:

$$\min_{\mathbf{S_x} \subseteq \mathbf{S}} \mathcal{Q}(\mathbf{Y}, f(\mathbf{S_x})) \tag{5.3}$$

where $\mathbf{S}$ is the set that includes all possible process variables, $\mathbf{S_x}$ is the subset of $\mathbf{S}$ that must be determined by the optimization problem, $f(\mathbf{S_x})$ gives the model prediction by using the variables from set $\mathbf{S_x}$, and $\mathcal{Q}(\cdot)$ represents benchmark function used to evaluate softsensor performance such as MSE or CC.

To obtain the optimal solution of Eqn. 5.3, $2^l - 1$ different scenarios must be considered theoretically, where l is the number of possible process variables. In practice, l is usually a large number that makes solving the problem very difficult. Nevertheless, by applying the stepwise regression method, the problem can be easily solved sub-optimally. At most, only $\frac{l^2+l}{2} - 1$ scenarios must be considered in the procedure of stepwise regression. For instance, if $l = 20$, i.e. 20 process variables need to be considered. One must try $2^{20} - 1 = 1,048,575$ cases in order to obtain the optimal solution. In contrast, using stepwise regression, only $\frac{20^2+20}{2} - 1 = 210$ cases need to be considered.

5.4 Process Description

In this application, softsensors for estimating pulp quality variables were developed. Pulp bleaching is a chemical process applied to cellulose material to increase its brightness, where brightness is defined as the reflectance of visible light from cellulose cloth or pulp fibres formed into sheets. Bleaching increases paper's capacity to accept printed or written images, and thus increases its usefulness. It is also a mean of purifying pulp - thereby extending its application, increasing its stability and enhancing some of its properties.

Chemicals commonly used for pulp bleaching include oxidants (e.g., chlorine, chlorine dioxide, oxygen, ozone, and hydrogen peroxide) and alkali (sodium hydroxide). These chemicals are mixed with pulp suspensions, and the mixture is retained at prescribed pH, temperature, and concentration conditions for a certain

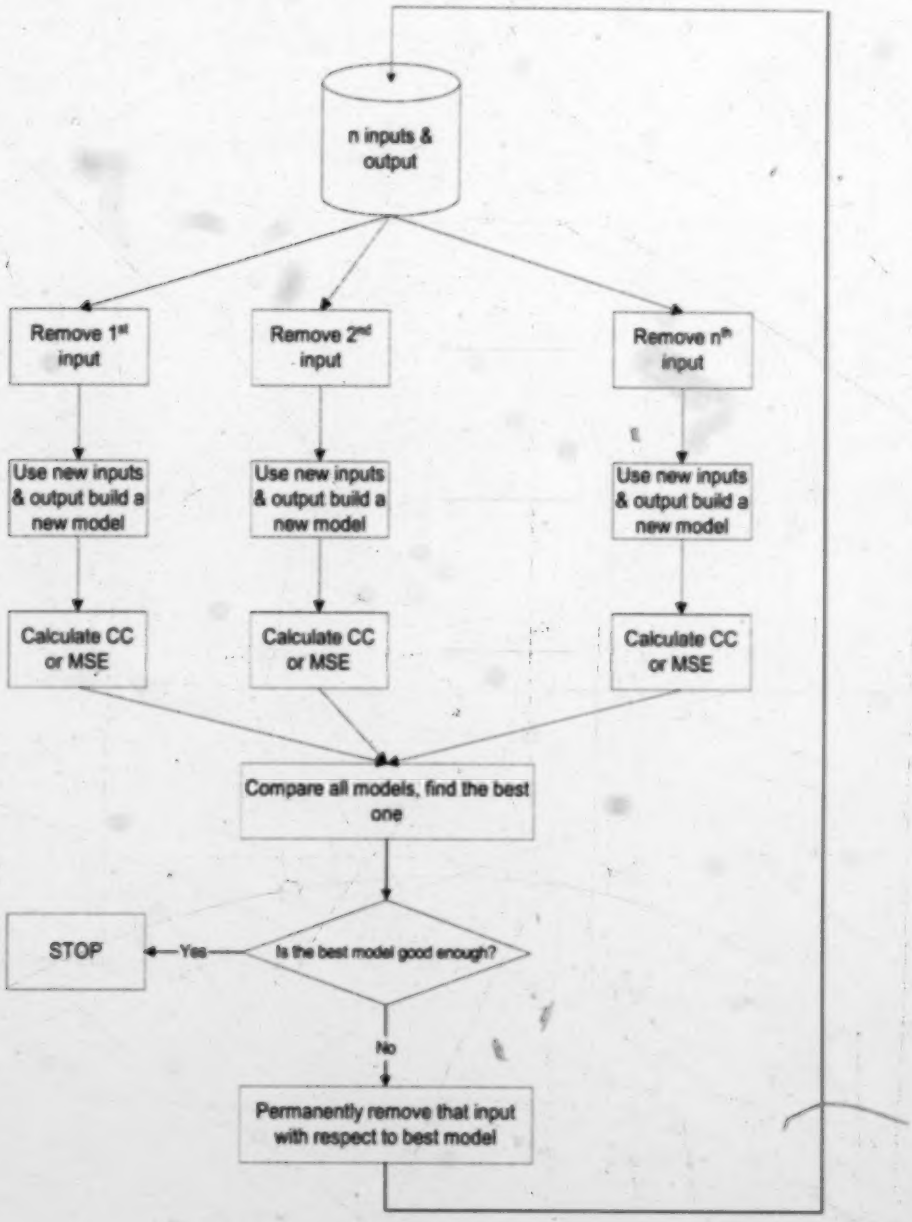
n inputs & output
Remove 1st input
Remove 2nd input
Remove nth input
Use new inputs & output build a new model
Use new inputs & output build a new model
Use new inputs & output build a new model
Calculate CC or MSE
Calculate CC or MSE
Calculate CC or MSE
Compare all models, find the best one
STOP
Yes
Is the best model good enough?
No
Permanently remove that input with respect to best model

Figure 5.2: Block diagram of "stepwise regression" method

time period. Due to the complexity of lignin and the wide variety of reactive bleaching species present, bleaching reactions are highly complex. The progress of bleaching reactions is monitored by measuring pulp lignin content, pulp brightness, and residual chemicals. Because it is not possible to achieve sufficient removal or decolorization of lignin by the action of any one chemical in a single treatment or stage, bleaching chemicals are frequently applied sequentially with intermediate washing between treatments (stages).

To carry out these reactions, appropriate process equipment is required for mixing steam with pulp to control the temperature, mixing chemicals with the pulp, pumping or otherwise conveying the pulp, and washing the pulp after the reaction is complete. Reaction times for bleaching are generally in the neighborhood of several hours, requiring the construction of large towers (reactors) to provide an adequate retention time.

Wood is the prime substance for making pulp, the raw material from which paper is manufactured. Wood is composed of cellulose fibres, with lignin holding the fibres together. It also contains resins, gums, and sugars. The fibres must be separated and arranged alone or with other materials to manufacture paper product. Pulp is made from the cellulose fibres of wood chips.

Millar Western employs a chlorine-free process that uses a combination of mild chemical, heat and mechanical action to produce bleached chemi-thermomechanical pulp (BCTMP). BCTMP is referred to as high-yield pulp, because the manufacturing process produces more pulp per tree than traditional pulping methods.

The bleaching process at Millar Western uses hydrogen peroxide as a bleaching agent. During the bleaching process, the cleaned and filtered pulp is squeezed in presses and heated before entering the bleaching tower P1, where it stays in a hydrogen peroxide solution for about 1.5 hours. The resulting semi-bleached pulp is dewatered in another press, and additional hydrogen peroxide is added in a chemical mixer. The second stage of bleaching takes about three and a half to five hours. The pulp is washed and pressed to extract bleach solution, which is recycled to the first stage of bleaching. Figure 5.3 illustrates the Millar Western BCTMP process. Millar Western produces more than 20 different specifications (grades) of hardwood, softwood and blended pulps, with a production capacity of up to 280,000 Air Dried Metric Tonnage (ADMT) of pulp per year, using approximately 800,000 cubic meters of timber and residual chips.

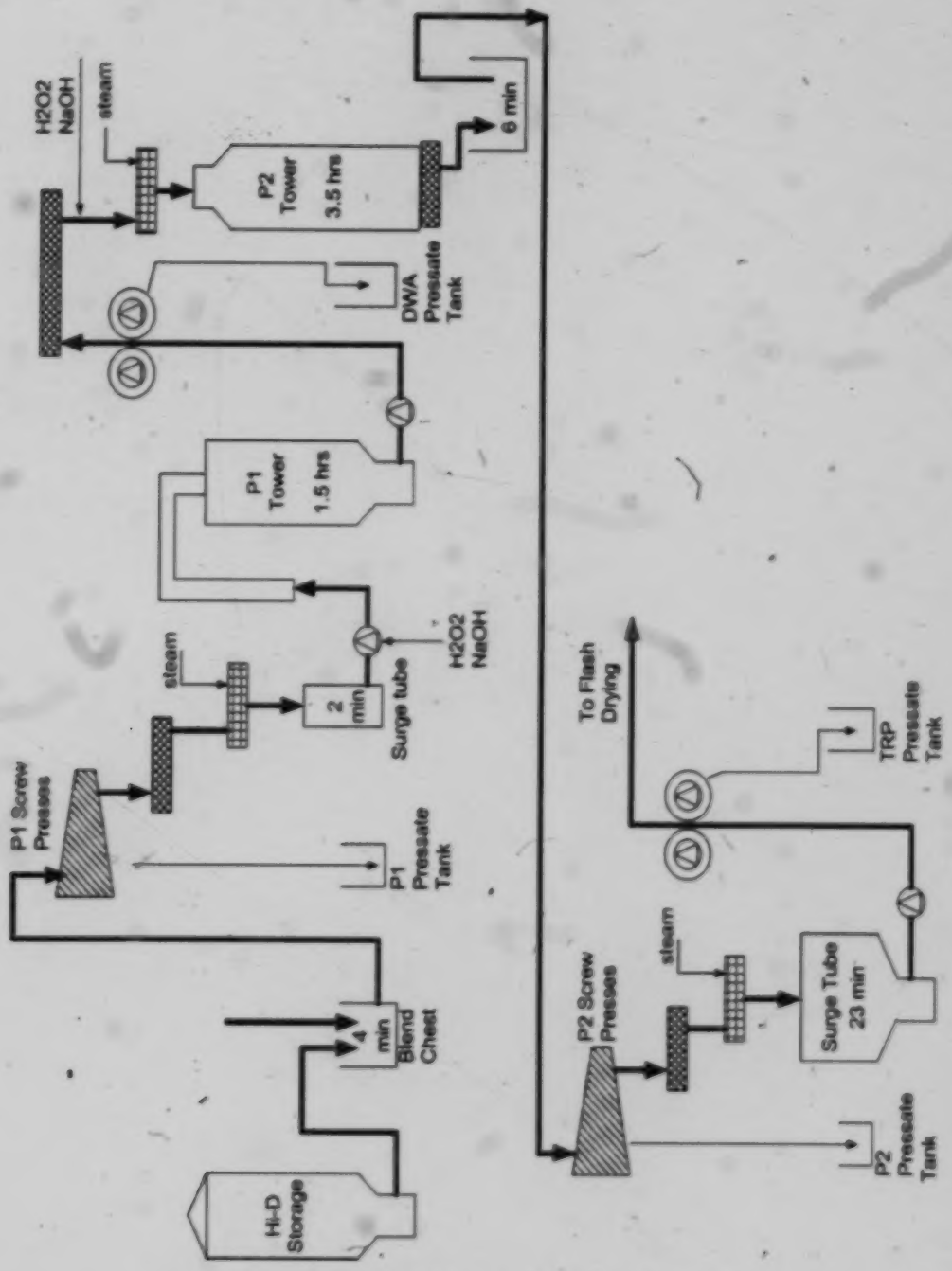

Figure 5.3: Millar Western Bleaching Process

5.5 Softsensor Development

5.5.1 Objective

Five process quality variables are deemed important in plant operation, and are therefore selected as the variables that must be estimated by softsensors. They are brightness, tensile strength, bulk density, opacity and scattering.

Building reliable softsensor models for this bleaching process is challenging, several difficulties must be overcome:

- **Little process knowledge:** Literature addressing bleaching process modelling is scarce. The first principle model, to my best knowledge, is not available yet.
- **Irregular output sampling:** All quality variables are sampled irregularly. The interval can be from 10 minutes to more than 20 hours. Generally, most of the intervals are approximately 4 hours.
- **Unknown nonlinear property:** From limited process knowledge, it has been observed that the bleaching process is obviously nonlinear. However, the type of nonlinearity is still unknown.

5.5.2 Preliminary Variable Selection

In total, 13 process variables are initially identified to be important from the process knowledge perspective. All of them are assumed to have effects on the product quality variables. However, some may be important and others may not under the normal operating conditions. The selected process variables (**U**) and their descriptions are listed in the Table 5.1.

The time delays of these variables are determined based on the retention times of P1 and P2 towers. However, the retention time is not always constant. It varies within a certain range. Even though retention time is not constant, using a constant time delay can dramatically simplify the problem of developing softsensor models. Therefore, in this study, the time delays shall be kept constant.

Due to the process property, i.e. time delay dominant system with little dynamic, it was decided to develop steady state softsensors using the PLS method. The detail of the PLS method can be found in the appendix. Using **U** and **Y** as formulated above, the PLS method is applied in order to develop quality variable softsensors. The first 2/3 of the data is used for model development, and the remaining 1/3 is

Table 5.1: Process variables

Name	Description
P1 PERO	PEROXIDE ADD RATE @ P1
P2 PERO	PEROXIDE ADD RATE @ P2
P1 CAUS	CAUSTIC ADD RATE @ P1
P2 CAUS	CAUSTIC ADD RATE @ P2
PROD RATE	PRODUCTION RATE
CONSIST	CONSISTENCY CORRECTION
P2 TEMP	P2 DISCHARGE TEMPERATURE
P2 DISC	P2 DISCHARGER CORRECTION
FLOW DIL	FLOW HMW TO BACKEND DIL
ASPEN	%ASPEN CHIPS
SULPHITE	Na_2SO_3 ADD ON CHIPS
CAUSTIC	CAUSTIC ADD ON CHIPS
PQM	PQM FREENESS

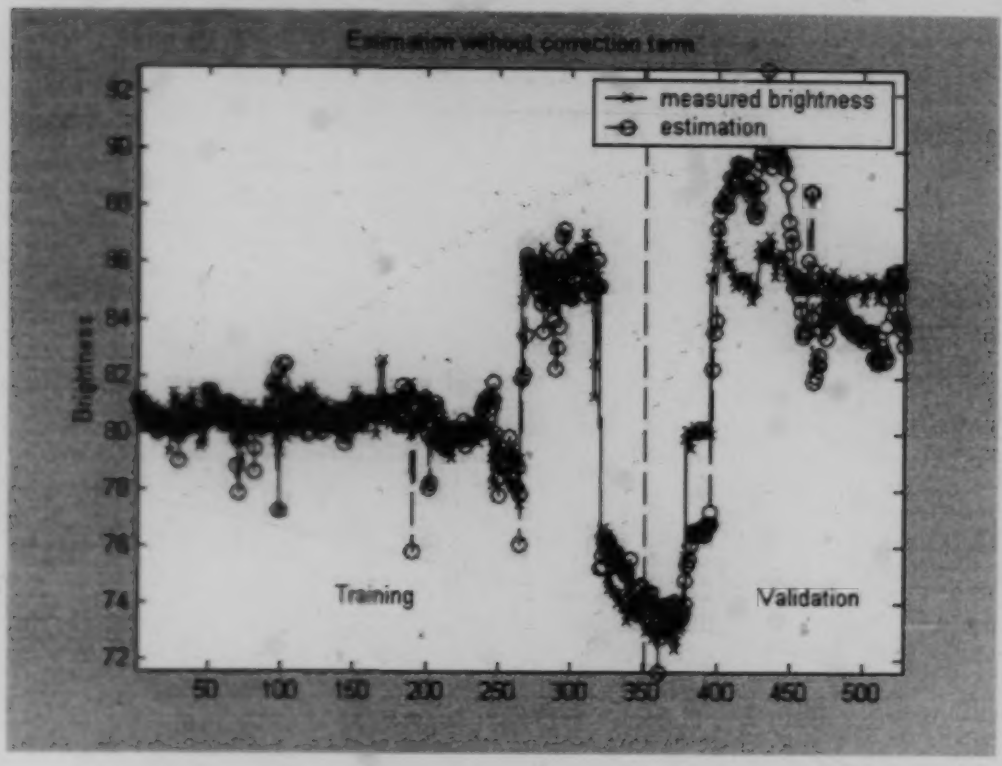

Figure 5.4: Preliminary results for Brightness versus time in sample periods

used for model validation. The preliminary result for brightness can be found in Figure 5.4. From this figure, it is obvious that the model does not have a satisfactory performance. More work must be done on this subject.

5.5.3 Nonlinear Consideration - Pseudo Varialbes

The bleaching process is known to be highly nonlinear due to the complex chemical reaction. The nonlinear property must be considered in this application. Because the linear modelling approach is being used here, one must create pseudo variables that are assumed to have a linear relationship to the quality variables. These pseudo variables are obtained based on the process knowledge. The pseudo variables in this study are listed in Table 5.2.

After considering these nonlinear properties of the process, a set of softsensors is developed based on the inputs - combining the original process variables and the

Table 5.2: Pseudo variables

Names	Descriptions
P2 RISE	TEMPERATURE RISE @ P2
TOTAL H2O2	TOTAL PEROXIDE
TOTAL NaOH	TOTAL CAUSTIC
P1 RATIO	CAUSTIC-PEROXIDE RATIO @ P1
P2 RATIO	CAUSTIC-PEROXIDE RATIO @ P2
ENERGY	SPECIFIC ENERGY

pseudo variables. The result for brightness softsensor is illustrated in Figure 5.5. The performance is dramatically increased compared to Figure 5.4.

5.5.4 Input Variable Selection

By applying the stepwise regression method to existing softsensor models, several interesting results can be obtained. In this study, MSE is used to evaluate model performance. Figure 5.6 shows the results for brightness.

The y-axis of Figure 5.6 is calculated based on the MSE value of the original model as follows:

$$\text{Performance}_i = \frac{\text{MSE}_i - \text{MSE}_{org}}{\text{MSE}_{org}} \times 100\% \tag{5.4}$$

where MSE_i denotes the MSE value of the model after i^{th} iteration and MSE_{org} represents the MSE value of the original model before performing stepwise regression. If "Performance_i" is negative, the new model has better performance than the original one. Otherwise, the new model is worse than the original.

The x-axis denotes the number of iterations, i.e. the number of inputs removed from the models. Since the y-axis and x-axis are defined in the above-mentioned way, the best model obtained during the iteration is the one with smallest "Performance_i" on the y-axis. This best model is referred as the "optimal model" in this chapter.

Table 5.3 lists the inputs selected by the "optimal model" for each softsensor. In this table, "o" means the corresponding "Tag Name" is selected based on stepwise regression; "×" denotes the input should be removed.

After selecting the input set of the "optimal" model and still using the PLS method, the softsensor models for each quality variable can be obtained. The result for

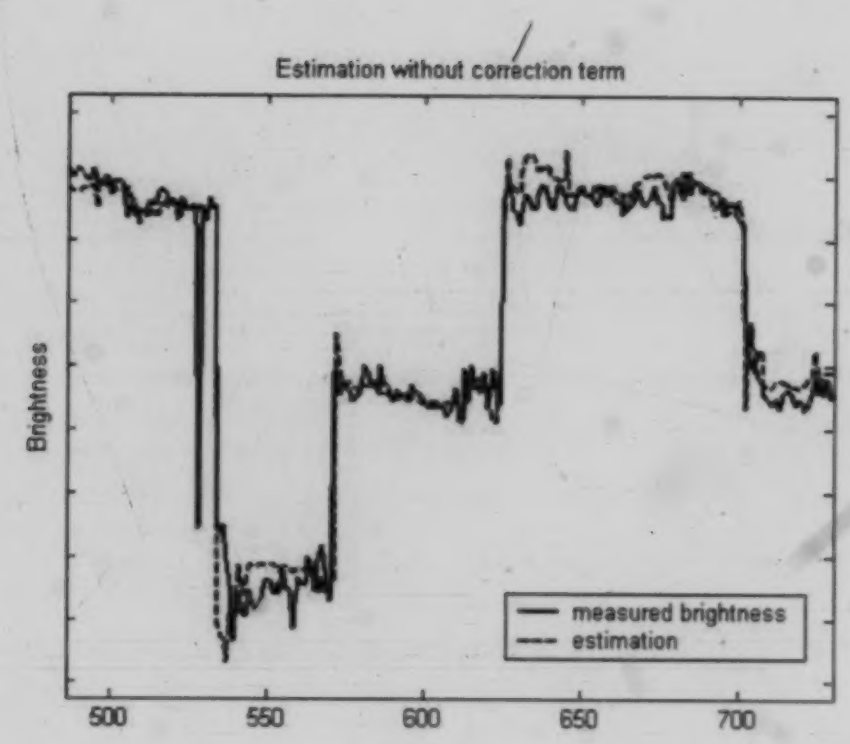

Figure 5.5: Brightness estimation with nonlinear consideration versus time in sampling periods

Table 5.3: Input variables selected for the "optimal models": o - selected; × - removed;

Names	Brightness	Tensile Strength	Bulk Density	Opacity	Scattering
P1 PERO	o	o	o	o	×
P2 PERO	o	o	o	o	×
P1 CAUS	o	o	o	o	o
P2 CAUS	o	o	o	o	×
PROD RATE	×	×	×	×	×
CONSIST	×	o	o	×	o
P2 TEMP	×	×	×	×	×
P2 DISC	o	o	o	×	×
FLOW DIL	×	o	o	×	×
ASPEN	o	×	×	×	×
SULPHITE	o	o	o	o	o
CAUSTIC	o	o	o	o	o
P2 RISE	o	×	o	o	×
PQM	×	×	×	o	o
TOTAL H2O2	o	o	o	o	×
TOTAL NaOH	o	o	o	o	o
P1 RATIO	o	o	o	o	o
P2 RATIO	o	×	o	×	×
ENERGY	×	×	×	×	×

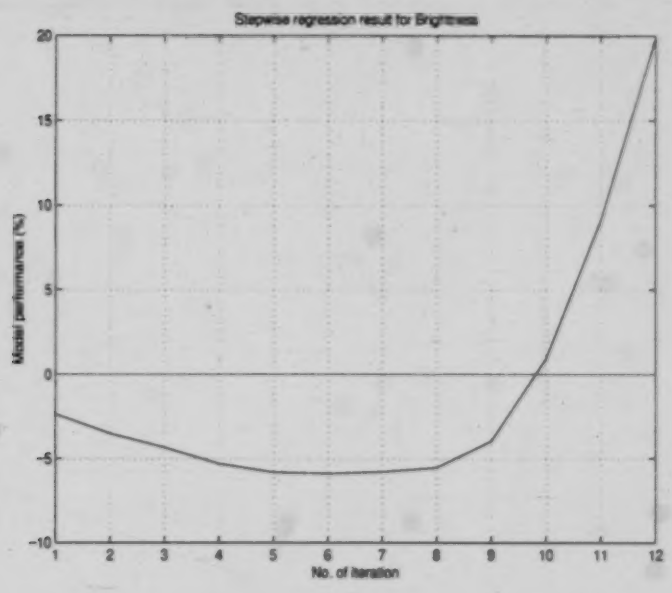

Figure 5.6: Stepwise regression results for Brightness

Table 5.4: Comparisons between the preliminary and the "optimal" model

Model	Brightness	Tensile	Density	Opacity	Scattering
CC (optimal)	0.9835	0.9612	0.9688	0.9777	0.9813
CC (preliminary)	0.9723	0.9345	0.9378	0.9742	0.9761
MSE (optimal)	0.0269	15.7620	0.0049	0.0717	0.1187
MSE (preliminary)	0.0687	41.5929	0.0148	0.1691	0.2874

brightness is shown in Figure 5.7.

Further, both CC and MSE are used to compare these new models with the results prior to stepwise regression in order to determine how the model quality improved after reducing the number of model inputs. Table 5.4 shows the comparisons.

From Table 5.4, it is obvious that the "optimal" model performs much better than the preliminary one regardless whether it is based on CC or MSE. At the same time, the "optimal" model has a simpler structure when compared to the preliminary one.

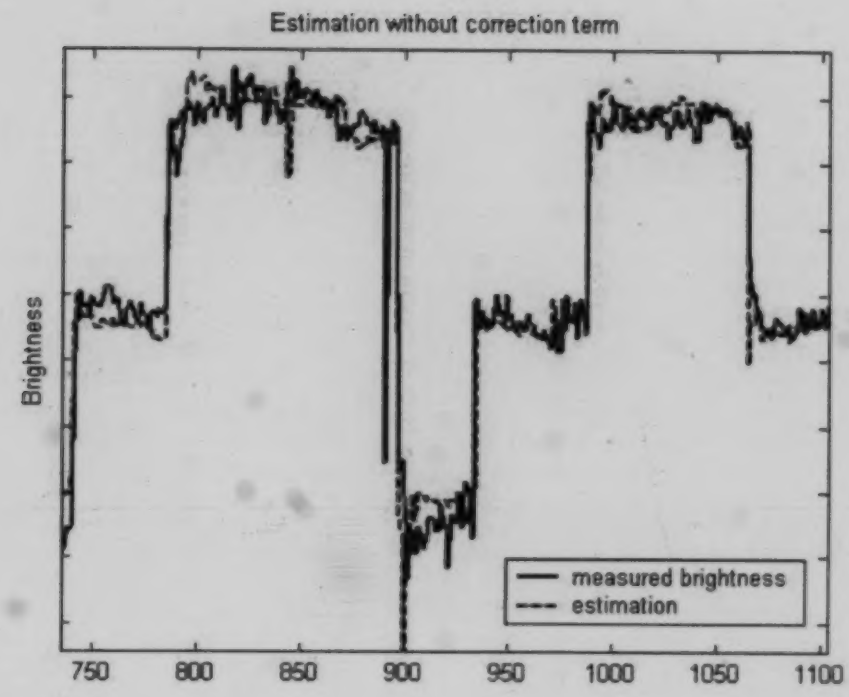

Figure 5.7: "Optimal" model result for Brightness

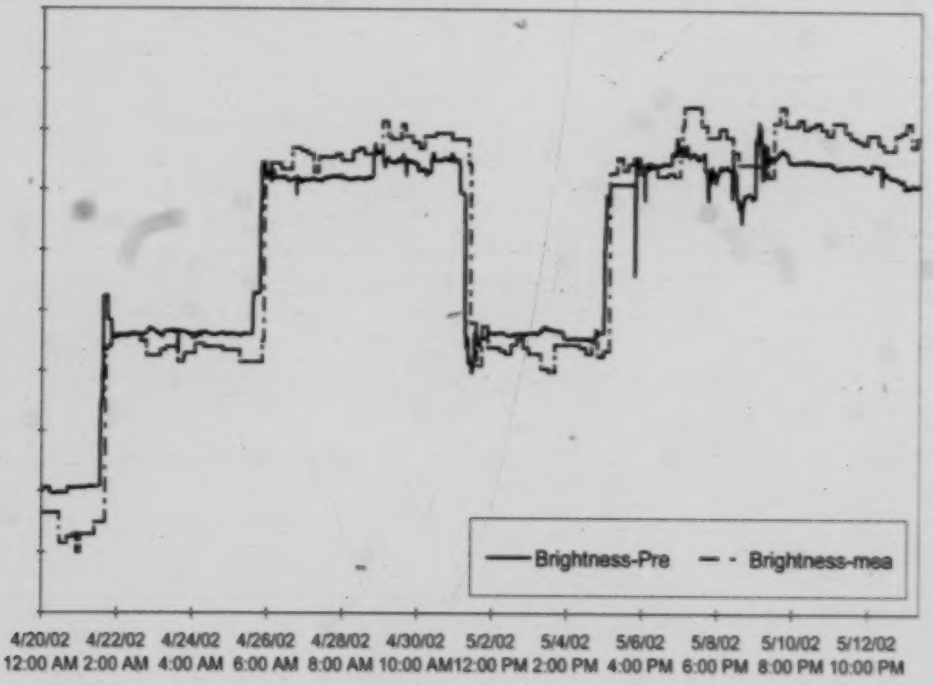

Figure 5.8: Online result for Brightness

5.5.5 Online Implementation and Results

After successfully designing the softsensor model, I put the "optimal" model online to evaluate its performance in practice. The implemented softsensors have been successfully running for over a year. Figure 5.8 illustrates the online brightness softsensor results from April 20 2002 to May 14 2002. The "optimal" model's performance is thus validated by these results. The softsensor models were running satisfactorily most of the time.

Note that all the results presented in this chapter are based on "feed forward" estimation, which is solely obtained from the process variable measurement. In contrast, "feed back" estimation uses the latest available laboratory measurements of the quality variables in addition to the process variable measurements to obtain the estimate. It is obvious that estimation with "feed back" tends to provide a more accurate performance. However, when evaluating model performance, it is preferable to use the "feed forward" approach. Doing so avoids the problem of lab

measurement masking the predication error, which makes it possible to assess the real model performance. Nevertheless, when the softsensor models are implemented in the real world, the "feed back" approach can be used to increase estimation accuracy.

5.6 Concluding Remarks

In this study, five softsensor models for estimating five different quality variables were successfully developed by using the PLS method. These softsensor models have been implemented online and validated in the real world. The stepwise regression method was proven to be useful for eliminating unimportant inputs for the purpose of improving model's overall performance. Nonlinear properties of the bleaching process at Millar Western were also considered in order to compensate for the nonlinear effect of the system.

The developed softsensors can be used for several purposes:

- *Process monitoring:* The softsensor gives the estimated inter-sample behavior of quality variables. Operators and control engineers can use this information to monitor the entire plant in operation. If the softsensor estimates is off target, one can infer that something in the process is not functioning properly.

- *Inferential control:* Because the softsensor provides the estimate at the same rate as the sampling rate of process variables, the softsensor estimate can be used as the feedback source to an advanced controller. Consequently, certain processes that were originally difficult to control due to measurement unavailability become controllable. Some applications have been made ([68]) in this area.

- *Process optimization:* Essentially, softsensor is a model that portrays the under relationship of the entire process. Not only does it identify the process variables that affect the process output - it also estimates the magnitude of the effects. One can then use this information to optimize the operation of the entire plant.

5.7 Appendix: the Partial Least Squares (PLS) Method

The PLS method is a well-known multivariate statistical analysis method that allows one to capture the maximum relationship between two data blocks: $\mathbf{Y} \in \Re^{N \times m}$ and $\mathbf{U} \in \Re^{N \times l}$, where N is the number of observations, m and l are the number of dependent and independent variables, respectively. The technical details of the PLS method are provided by Geladi and Kowalski (1986, [35]), and Höskuldsson (1988, [51]). Only a brief introduction of this method is presented here.

In a linear steady state process, $\mathbf{Y}$ and $\mathbf{U}$ can be related by the following linear equation:

$$\mathbf{Y} = \mathbf{U}\boldsymbol{\theta} + \tilde{\mathbf{E}} \tag{5.5}$$

where $\boldsymbol{\theta} \in \Re^{l \times m}$ is the model parameter, and $\tilde{\mathbf{E}} \in \Re^{N \times m}$ is the process noise.

According to the ordinary least squares (OLS) theory, if the matrix $\mathbf{U}'\mathbf{U}$ is non-singular, the best estimation in the least squares sense, provided $\tilde{\mathbf{E}}$ is a zero-mean white noise process, is: $\hat{\boldsymbol{\theta}} = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{Y}$. However, in practice, the matrix $\mathbf{U}$ is most unlikely to be of full rank or $\mathbf{U}'\mathbf{U}$ is unlikely to be well conditioned. In order to solve this problem, the PLS method was proposed to find the $\hat{\boldsymbol{\theta}}$ which can provide the maximum covariance matrix between linear combinations of $\mathbf{Y}$ and $\mathbf{U}$. The problem can be described in the optimization framework as follows:

$$\begin{aligned} \max_{\mathbf{w}_1, \mathbf{q}_1} cov(\mathbf{t}_1, \mathbf{h}_1) &= \mathbf{t}_1'\mathbf{h}_1 = \mathbf{w}_1'\mathbf{U}'\mathbf{Y}\mathbf{q}_1 \\ s.t.\ \ \mathbf{w}_1'\mathbf{w}_1 &= 1 \\ \mathbf{q}_1'\mathbf{q}_1 &= 1 \end{aligned} \tag{5.6}$$

where $\mathbf{t}_1 = \mathbf{U}\mathbf{w}_1$ and $\mathbf{h}_1 = \mathbf{Y}\mathbf{q}_1$ are linear combinations of $\mathbf{U}$ and $\mathbf{Y}$ respectively. $cov(\mathbf{a}, \mathbf{b})$ stands for the covariance between two vectors $\mathbf{a}$ and $\mathbf{b}$.

Using Lagrangian multiplier and singular value decomposition (SVD), $\mathbf{w}_1$ and $\mathbf{q}_1$ can be proven to be the first column of matrix $\mathbf{L}$ and $\mathbf{V}$ respectively, where $\mathbf{U}'\mathbf{Y} = \mathbf{L}\mathbf{S}\mathbf{V}'$ (Höskuldsson 1988, [51]; Manne 1987, [82]). Once $\mathbf{t}_1$ and $\mathbf{h}_1$ are obtained via $\mathbf{t}_1 = \mathbf{U}\mathbf{w}_1$ and $\mathbf{h}_1 = \mathbf{Y}\mathbf{q}_1$, a linear relation between $\mathbf{h}_1$ and $\mathbf{t}_1$ can be found to be: $\hat{\mathbf{h}}_1 = b_1\mathbf{t}_1$, where $b_1 = \dfrac{\mathbf{h}_1'\mathbf{t}_1}{\mathbf{t}_1'\mathbf{t}_1}$ is the linear regression coefficient. Thus the data blocks $\mathbf{U}$ and $\mathbf{Y}$ are indirectly related through the inner relation between $\mathbf{h}_1$ and $\mathbf{t}_1$.

Therefore, the remaining unexplained information in matrices $\mathbf{U}$ and $\mathbf{Y}$ can be calculated as follows:

$$\mathbf{E}_1 = \mathbf{U} - \mathbf{t}_1\mathbf{p}_1', \quad \mathbf{F}_1 = \mathbf{Y} - b_1\mathbf{t}_1\mathbf{q}_1' \tag{5.7}$$

where $\mathbf{p}_1 = \dfrac{\mathbf{U}'\mathbf{t}_1}{\mathbf{t}_1'\mathbf{t}_1}$ is to ensure that the next score vector is orthogonal to the previous one. The above procedure is repeated until the useful information in matrix $\mathbf{Y}$ can not be further explained by adding any more latent variables. Choosing a proper number of the latent variables n_a is a crucial issue in PLS. Wold (1978, [132]) suggests cross validation method to determine the proper n_a. In addition, Denham (2000, [24]) investigated the determination of the number of factors in PLS framework.

From a practical point of view, PLS can be considered as a technique that breaks up a multivariate regression problem into a series of univariate regression problems. The original regression problem is handled by constructing n_a inner relationship models (usually $n_a \ll n$). In matrix form, the prediction model can be expressed as:

$$\hat{\mathbf{Y}} = b_1\mathbf{t}_1\mathbf{q}_1' + \cdots + b_{n_a}\mathbf{t}_{n_a}\mathbf{q}_{n_a}' = \mathbf{TBQ}' \tag{5.8}$$

where $\hat{\mathbf{Y}}$ is the predictive output, $\mathbf{B}$ is a diagonal matrix with $b_1,\ b_2,\ \cdots,\ b_{n_a}$ on the diagonal, $\mathbf{T} = [\mathbf{t}_1,\ \mathbf{t}_2,\ \cdots,\ \mathbf{t}_{n_a}]$ and $\mathbf{Q} = [\mathbf{q}_1,\ \mathbf{q}_2,\ \cdots,\ \mathbf{q}_{n_a}]$.

The iterative algorithm of PLS was developed by Höskuldsson (1988), which can be briefly stated as follows:

1. Set $\mathbf{y}$ to a column of $\mathbf{Y}$
2. Regress columns of $\mathbf{U}$ on $\mathbf{y}$: $\mathbf{c} = \mathbf{U}'\mathbf{y}/\|\mathbf{y}\|^2$
3. Normalize $\mathbf{c}$ to unit length: $\mathbf{c} = \mathbf{c}/\|\mathbf{c}\|^2$
4. Calculate the scores: $\mathbf{t} = \mathbf{Uc}$
5. Regress columns of $\mathbf{Y}$ on $\mathbf{t}$: $\mathbf{g} = \mathbf{Y}'\mathbf{t}/\|\mathbf{t}\|^2$
6. Normalize $\mathbf{g}$ to unit length: $\mathbf{g} = \mathbf{g}/\|\mathbf{g}\|^2$
7. Calculate new $\mathbf{y}$ vector: $\mathbf{y} = \mathbf{Yg}$
8. Check the convergence: if $\|\mathbf{y} - \mathbf{y}_{old}\| \leq$ convergence criterion to step 9; if not to step 2
9. $\mathbf{U}$ loadings: $\mathbf{w} = \mathbf{U}'\mathbf{t}/\|\mathbf{t}\|^2$

10. $\mathbf{Y}$ loadings: $\mathbf{q} = \mathbf{Y}'\mathbf{y}/\|\mathbf{y}\|^2$

11. Regression: $b = \mathbf{y}'\mathbf{t}$

12. Deflation: $\mathbf{U} = \mathbf{U} - \mathbf{t}\mathbf{w}'$ and $\mathbf{Y} = \mathbf{Y} - b\mathbf{t}\mathbf{q}'$

13. To calculate the next set of latent vectors repeat; else, stop.

In addition, Kaspar and Ray (1992, [60]) give an excellent graphical interpretation of PLS, which makes PLS very understandable.

6

Canonical Variate Analysis for Ill-conditioned Data[3]

6.1 Introduction

An Increase in the complexity and instrumentation of industrial processes has resulted in a proliferation of data consisting of large numbers of highly correlated variables. Examples of this activity include the fields of organic and analytical chemistry, process control, biotechnology, food science, pharmacology and environmental research ([14]). Sometimes, one is interested in uncovering, describing and summarizing structural relationships among the measured variables. However, in most applications, given a set of process variables, the objective is to relate a subset of variables, termed

[3]A version of this chapter has been submitted for possible publication to the *Journal of Process Control* by Li, Han and Shah.

the *response variables* or the *outputs*, to the remaining process variables, termed the *explanatory variables* or the *inputs*. This procedure is well known as *multivariate regression*. For instance, one may be interested in relating a set of quality variables of a product to another set of variables in the associated production process. Due to the high degree of correlation among variables in both the input and output spaces, multivariate regression techniques based on latent variables such as canonical variate analysis (CVA) (also termed canonical correlation analysis (CCA) in some literatures), principal component regression (PCR), and partial least squares (PLS) have been sought for this purpose ([14] [60] [128]).

Similarities and differences exist among various latent variables-based regression techniques. They are similar in that each of them uses latent variables transformed from the inputs, instead of the inputs themselves, to regress the outputs. However, each technique constructs its latent variables in a unique manner. While PCR is most effective at removing redundant variables in the inputs, CVA is the best at relating the inputs to the outputs. PLS is somewhere in between PCR and CVA, as graphically illustrated by Kaspar and Ray (1992, [60]). Furthermore, geometrically, the goal of CVA is to minimize the angle between each pair of latent vectors ([64]), which are transformed from the input and output data respectively. Nevertheless, PLS pursues minimization of the distance between each pair of latent vectors ([51]).

CVA has been extended to dynamic systems as well. For instance, Larimore ([67] [66]) has developed the CVA-based subspace method of identification (SMI) for state space models of processes. Furthermore, the CVA-based SMI has been applied to process monitoring and fault detection ([117] [89]). Like PCR and PLS, CVA transforms the original inputs and outputs into two sets of *latent variables*. The latent variables can also be defined as *canonical variables* or *scores*. Subsequently, regression is performed between two latent variables with each pair. Every latent variable is a linear combination of its associated variables.

However, existing CVA algorithms have a fundamental weakness. Notice that in CVA, each latent variable is scaled to have unit variance. Because a latent variable is simply a linear combination of the original process variables, in order to ensure that each latent variable is of unit variance one must scale the original process variables by the square root of their covariance matrix. If collinearity exists in the inputs and/or outputs, process data are referred to as *ill-conditioned*. With such data, the covariance matrix of the original process variables has null eigenvalues. As a result, the inverse of the covariance does not exist, making the afore-mentioned scaling impossible. Because it fails to deal with collinearity either in the input and/or output variables, strictly

speaking, in our viewpoint, existing CVA methodology cannot be categorized as a latent variable-based regression approach, while PCR and PLS definitely can.

This chapter proposes a novel CVA algorithm that is insensitive to ill-conditioned data while preserving all the advantages of conventional CVA algorithms. To treat the singularity in the covariance matrices of the input and/or output data, a truncated Cayley Hamilton series must first be employed to best approximate the inverse of the covariance matrix. Then, the model parameter matrix, which associates the inputs with the outputs, is estimated in terms of a reduced Krylov controllability matrix. The latent variables for the inputs and outputs can be directly calculated from the data and the Krylov matrix without calculating the weighting vectors.

Although Di Ruscio (2000, [113]) has pioneered the use of the above-mentioned reduced Krylov matrix for PLS regression, extension of the concept to CVA is a major contribution of this chapter.

6.2 Numerical Problem of Existing CVA

Why does a conventional CVA algorithm not work with ill-conditioned data? To answer this question in detail, let us first revisit the CVA-based multivariate regression method.

6.2.1 CVA-based Multivariate Regression

It is assumed that the process can be represented by the following regression model:

$$\tilde{\mathbf{y}}(k) = \mathbf{M}\tilde{\mathbf{u}}(k_d) \tag{6.1}$$

Assume that a series of data $\{\mathbf{u}(k), \mathbf{y}(k)\}$ is collected from the above-mentioned process. If both $\mathbf{u}(k)$ and $\mathbf{y}(k)$ are corrupted by noise, one can represent them as follows:

$$\mathbf{u}(k) = \tilde{\mathbf{u}}(k) + \mathbf{v}(k), \quad \mathbf{y}(k) = \tilde{\mathbf{y}}(k) + \mathbf{o}(k) \tag{6.2}$$

Since only $\mathbf{u}(k)$ and $\mathbf{y}(k)$ are available, use of Eqn. 6.2 in Eqn. 6.1 gives:

$$\mathbf{y}(k) = \mathbf{M}\mathbf{u}(k_d) + \mathbf{e}(k) \tag{6.3}$$

where $\mathbf{e}(k) = \mathbf{o}(k) - \mathbf{M}\mathbf{v}(k)$ accounts for the effects of measurement noise in the inputs and outputs. From the assumed distributions of $\mathbf{v}(k)$ and $\mathbf{o}(k)$, it can be

inferred that $\mathbf{e}(k)$ follows a multivariate Gaussian distribution with zero mean and a covariance of $\mathbf{R}_e = \mathbf{R}_o + \mathbf{M}\mathbf{R}_v\mathbf{M}' \in \Re^{m\times m}$ ([58]).

Using N samples of collected data, one can designate one input block $\mathbf{U} \in \Re^{N\times l}$ and one output block $\mathbf{Y} \in \Re^{N\times m}$. In addition, we define another data block as $\mathbf{E} \in \Re^{N\times m}$ for $\mathbf{e}(k)$, which has a format similar to that of $\mathbf{U}$ and $\mathbf{Y}$. In each block, the rows correspond to *samples*, while the columns correspond to the *variables*. With these data blocks, one can transform Eqn. 6.3 into:

$$\mathbf{Y} = \mathbf{U}\mathbf{M}' + \mathbf{E} \tag{6.4}$$

CVA does not use $\mathbf{U}$ to regress $\mathbf{Y}$ directly. Instead, the objective of CVA is to find latent variable vectors $\mathbf{t}_j = \mathbf{U}\mathbf{p}_j$ from the column spaces of $\mathbf{U}$ to regress $\mathbf{Y}$, or to regress the latent vectors $\mathbf{h_j} = \mathbf{Y}\mathbf{w}_j$ of $\mathbf{Y}$. Like their counterparts in PLS, $\mathbf{p}_j$ and $\mathbf{w}_j$ are defined as the j^{th} *weight vectors* of $\mathbf{U}$ and $\mathbf{Y}$ respectively. However, unlike PLS, the solution to $\mathbf{p}_j$ and $\mathbf{w}_j$ must be such that the resulting $\mathbf{t}_j$ and $\mathbf{h}_j$ are of highest correlation.

To calculate the first pair of latent variable vectors $\{\mathbf{t}_1, \mathbf{h}_1\}$, the following objective function can be established:

$$\max_{\mathbf{p}_1,\mathbf{w}_1} (\mathbf{U}\mathbf{p}_1)'\mathbf{Y}\mathbf{w}_1 \tag{6.5}$$

subject to $(\mathbf{U}\mathbf{p}_1)'\mathbf{U}\mathbf{p}_1 = 1$ and $(\mathbf{Y}\mathbf{w}_1)'\mathbf{Y}\mathbf{w}_1 = 1$.

Further, the successive pairs of latent variable vectors $\{\mathbf{t}_j, \mathbf{h}_j\}$ are calculated by maximizing

$$(\mathbf{U}\mathbf{p}_j)'\mathbf{Y}\mathbf{w}_j \tag{6.6}$$

under these constraints:

$$(\mathbf{U}\mathbf{p}_j)'\mathbf{U}\mathbf{p}_j = 1,\ \ (\mathbf{Y}\mathbf{w}_j)'\mathbf{Y}\mathbf{w}_j = 1,$$

$$(\mathbf{U}\mathbf{p}_j)'\mathbf{U}\mathbf{p}_i = 0,\ \ (\mathbf{Y}\mathbf{w}_j)'\mathbf{Y}\mathbf{w}_i = 0,\ \ (\mathbf{U}\mathbf{p}_j)'\mathbf{Y}\mathbf{w}_i = 0$$

where $i \neq j$, $1 \leq i \leq \min\{l, m\}$ and $2 \leq j \leq \min\{l, m\}$.

Suppose that $1 \leq n_\circ \leq \min\{l, m\}$ latent variables have been selected from $\mathbf{U}$. It follows from Burnham et al. (1996, [14]) that the estimate of $\mathbf{M}$ in terms of $n_\circ$ latent variables of $\mathbf{U}$ is

$$\hat{\mathbf{M}}' = \mathbf{P}_{n_\circ}\mathbf{T}'_{n_\circ}\mathbf{Y} \tag{6.7}$$

where $\mathbf{P}_{n_\circ} = [\mathbf{p}_1 \cdots \mathbf{p}_{n_\circ}] \in \Re^{M \times n_\circ}$ are $n_\circ$ weight vectors for $\mathbf{U}$, and $\mathbf{T}_{n_\circ} = [\mathbf{t}_1 \cdots \mathbf{t}_{n_\circ}] \in \Re^{N \times n_\circ}$ are the associated latent variable vectors.

Applying Eqn. 6.2 to Eqn. 6.7, the predicted value $\hat{\mathbf{y}}(k)$ of $\mathbf{y}(k)$ is as follows:

$$\hat{\mathbf{y}}(k) = \mathbf{Y}'\mathbf{T}_{n_\circ}\mathbf{P}'_{n_\circ}\mathbf{u}(k) = \mathbf{Y}'\mathbf{T}_{n_\circ}\mathbf{t}(k) \tag{6.8}$$

where $\mathbf{t}(k) = \mathbf{P}'_{n_\circ}\mathbf{u}(k) \in \Re^{n_\circ}$ are the latent variables of $\mathbf{u}(k)$. Moreover, the prediction error of $\mathbf{y}(k)$ is:

$$\tilde{\mathbf{y}}(k) \equiv \mathbf{y}(k) - \hat{\mathbf{y}}(k) = \mathbf{y}(k) - \mathbf{Y}'\mathbf{T}_{n_\circ}\mathbf{t}(k) \tag{6.9}$$

Finally, pre-multiplying Eqn. 6.9 by $\mathbf{W}'_{n_\circ}$ shows that

$$\tilde{\mathbf{h}}(k) \equiv \mathbf{h}(k) - \hat{\mathbf{h}}(k) = \mathbf{h}(k) - \mathbf{H}'_{n_\circ}\mathbf{T}_{n_\circ}\mathbf{t}(k) \tag{6.10}$$

where $\mathbf{H}_{n_\circ} = \mathbf{Y}\mathbf{W}_{n_\circ} = [\mathbf{h}_1 \cdots \mathbf{h}_{n_\circ}] \in \Re^{N \times n_\circ}$ are $n_\circ$ latent variable vectors of $\mathbf{Y}$, $\mathbf{h}(k) = \mathbf{W}'_{n_\circ}\mathbf{y}(k) \in \Re^{n_\circ}$ are latent variables of $\mathbf{y}(k)$, and $\mathbf{W}_{n_\circ} = [\mathbf{w}_1 \cdots \mathbf{w}_{n_\circ}] \in \Re^{m \times n_\circ}$ are $n_\circ$ weight vectors for $\mathbf{Y}$. Note that $\mathbf{H}'_{n_\circ}\mathbf{T}_{n_\circ} \in \Re^{n_\circ \times n_\circ}$ is a diagonal matrix due to the constraints imposed in Eqns. 6.5 and 6.6. Denoting $\mathbf{\Omega}_{n_\circ} \equiv \mathbf{H}'_{n_\circ}\mathbf{T}_{n_\circ}$, one can rewrite Eqn. 6.10 as

$$\tilde{\mathbf{h}}(k) \equiv \mathbf{h}(k) - \hat{\mathbf{h}}(k) = \mathbf{h}(k) - \mathbf{\Omega}_{n_\circ}\mathbf{t}(k) \tag{6.11}$$

which is the CVA-based regression model of the process under consideration in terms of latent variables.

6.2.2 Numerical Problem with Ill-conditioned Data

As derived by Johnson and Wichern ([58]), $\{\mathbf{p}_1, \cdots, \mathbf{p}_j\}$ and $\{\mathbf{w}_1, \cdots, \mathbf{w}_j\}$ are the eigenvectors corresponding to j largest eigenvalues of matrices $(\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{Y}(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{U}$ and $(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{U}(\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{Y}$, respectively. That is

$$(\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{Y}(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{U}\mathbf{p}_j = \lambda_j\mathbf{p}_j \tag{6.12}$$

and

$$(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{U}(\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{Y}\mathbf{w}_j = \lambda_j\mathbf{w}_j \tag{6.13}$$

for $1 \leq j \leq n_\circ$. The following remarks can be made on the basis of Eqns. 6.12 and 6.13.

Remark 6.2.1 The non-trivial solutions to $\mathbf{p}_j$ and $\mathbf{w}_j$ exist only if $\mathbf{U}'\mathbf{U}$ and $\mathbf{Y}'\mathbf{Y}$ are non-singular.

Remark 6.2.2 If $\mathbf{U}'\mathbf{U}$ and $\mathbf{Y}'\mathbf{Y}$ are not singular but have very small eigenvalues, numerical problems may arise in calculating $\mathbf{p}_j$ and $\mathbf{w}_j$. In this case, even slight perturbations in the eigenstructures of $\mathbf{U}'\mathbf{U}$ and/or $\mathbf{Y}'\mathbf{Y}$ can cause substantial variations in $\mathbf{p}_j$ and/or $\mathbf{w}_j$. For example, a serious mismatch may exist between the earlier estimated process model $\hat{\mathbf{M}}$ and the newly sampled data, even in the presence of normal uncertainties in the process.

Remark 6.2.3 Due to the inevitability of measurement noise in $\{\mathbf{y}(k)\}$, the non-singularity of $\mathbf{Y}'\mathbf{Y}$ can be ensured in most cases, except in the extreme case where the number of samples is smaller than the number of variables, i.e. $N < m$. As a result, one does not need worry about the existence of $(\mathbf{Y}'\mathbf{Y})^{-1}$. However, one may have to concern the singularity of $\mathbf{U}'\mathbf{U}$. In a closed loop control system, $\mathbf{u}(k) \in \Re^l$ are the outputs of controllers. If $\mathbf{u}(k)$ are measured, the contribution of noise can guarantee the non-singularity of $\mathbf{U}'\mathbf{U}$. If $\mathbf{u}(k)$ are calculated, then $\mathbf{U}'\mathbf{U}$ may be singular. The key to overcoming the weakness of existing CVA algorithms is to approximate the inverse of $\mathbf{U}'\mathbf{U}$ using an optimal reduced Krylov matrix. This will result in the development of ill-conditioned data insensitive CVA.

6.3 CVA Insensitive to Ill-conditioned Data

In this section, first it will be illustrated that the estimate $\hat{\mathbf{M}}$ of $\mathbf{M}$ matrix in Eqn. 6.4 is a function of a Krylov controllability matrix. Subsequently, in the presence of singularity in $\mathbf{U}'\mathbf{U}$, optimal approximation of $(\mathbf{U}'\mathbf{U})^{-1}$ is investigated. Finally, a CVA insensitive to ill-conditioned data is developed by analyzing the structure of the Krylov controllability matrix.

6.3.1 Reduced Krylov Controllability Matrix

In order to introduce the concept of the Krylov controllability matrix, we will assume that at this moment $\mathbf{U}'\mathbf{U}$ is not singular. Consequently, it is trivial to derive the least squares (LS) estimate of $\mathbf{M}$ as below:

$$\hat{\mathbf{M}}' = (\mathbf{U}'\mathbf{U})^{-1}\mathbf{U}'\mathbf{Y} \tag{6.14}$$

Assume further that the characteristic polynomial of matrix $\mathbf{U}'\mathbf{U}$ is

$$Det\left(\lambda \mathbf{I}_l - \mathbf{U}'\mathbf{U}\right) = \lambda^l + \alpha_1 \lambda^{l-1} + \cdots + \alpha_{l-1}\lambda + \alpha_l = 0,$$

where $Det(\cdot)$ represents the determinant of a matrix, $\lambda > 0$ is an eigenvalue of the matrix $\mathbf{U}'\mathbf{U}$, and $\alpha_j \neq 0$ ($j \in [1, l]$) is the j^{th} coefficient of the polynomial. Consequently, from the well-known Cayley-Hamilton Theorem it follows that

$$\left(\mathbf{U}'\mathbf{U}\right)^l + \alpha_1 \left(\mathbf{U}'\mathbf{U}\right)^{l-1} + \cdots + \alpha_{l-1}\left(\mathbf{U}'\mathbf{U}\right) + \alpha_l \mathbf{I}_l = 0 \tag{6.15}$$

Based on Eqn. 6.15, the following can be derived:

$$\begin{aligned} \mathbf{I}_l &= -\frac{1}{\alpha_l}\left(\left(\mathbf{U}'\mathbf{U}\right)^l + \alpha_1 \left(\mathbf{U}'\mathbf{U}\right)^{l-1} + \cdots + \alpha_{l-1}\left(\mathbf{U}'\mathbf{U}\right)\right) \\ &= \left[\left(\mathbf{U}'\mathbf{U}\right)^l \quad \left(\mathbf{U}'\mathbf{U}\right)^{l-1} \quad \cdots \quad \mathbf{U}'\mathbf{U}\right] \begin{bmatrix} \alpha_1^* \mathbf{I}_l \\ \vdots \\ \alpha_l^* \mathbf{I}_l \end{bmatrix} \end{aligned} \tag{6.16}$$

where α_i^* is the i^{th} element of vector $\boldsymbol{\psi}_l = -\frac{1}{\alpha_l}\left[1 \;\; \alpha_1 \;\; \cdots \;\; \alpha_{l-2} \;\; \alpha_{l-1}\right]' \in \Re^l$ for $i \in [1, l]$. Hence, multiplying both sides of Eqn. 6.16 by $\left(\mathbf{U}'\mathbf{U}\right)^{-1}$ leads to

$$\left(\mathbf{U}'\mathbf{U}\right)^{-1} = \left[\left(\mathbf{U}'\mathbf{U}\right)^{l-1} \quad \left(\mathbf{U}'\mathbf{U}\right)^{l-2} \quad \cdots \quad \mathbf{U}'\mathbf{U} \quad \mathbf{I}_l\right] \begin{bmatrix} \alpha_1^* \mathbf{I}_l \\ \vdots \\ \alpha_l^* \mathbf{I}_l \end{bmatrix} \tag{6.17}$$

Finally, applying Eqn. 6.17 to Eqn. 6.14 results in:

$$\begin{aligned} \hat{\mathbf{M}}' &= \left[\left(\mathbf{U}'\mathbf{U}\right)^{l-1} \quad \left(\mathbf{U}'\mathbf{U}\right)^{l-2} \quad \cdots \quad \mathbf{U}'\mathbf{U} \quad \mathbf{I}_l\right] \begin{bmatrix} \alpha_1^* \mathbf{I}_l \\ \vdots \\ \alpha_l^* \mathbf{I}_l \end{bmatrix} \mathbf{U}'\mathbf{Y} \\ &= \left[\left(\mathbf{U}'\mathbf{U}\right)^{l-1}\mathbf{U}'\mathbf{Y} \quad \cdots \quad \mathbf{U}'\mathbf{U}\mathbf{U}'\mathbf{Y} \quad \mathbf{U}'\mathbf{Y}\right] \begin{bmatrix} \alpha_1^* \mathbf{I}_m \\ \vdots \\ \alpha_l^* \mathbf{I}_m \end{bmatrix} \end{aligned} \tag{6.18}$$

Denote

$$\mathbf{K}_l \equiv \left[\left(\mathbf{U}'\mathbf{U}\right)^{l-1}\mathbf{U}'\mathbf{Y} \quad \left(\mathbf{U}'\mathbf{U}\right)^{l-2}\mathbf{U}'\mathbf{Y} \quad \cdots \quad \mathbf{U}'\mathbf{U}\mathbf{U}'\mathbf{Y} \quad \mathbf{U}'\mathbf{Y}\right] \in \Re^{l \times (lm)}$$

With such a definition, Eqn. 6.18 can be rewritten as

$$\hat{\mathbf{M}}' \equiv \mathbf{K}_l \begin{bmatrix} \alpha_1^* \mathbf{I}_m \\ \vdots \\ \alpha_l^* \mathbf{I}_m \end{bmatrix}$$

indicating that the LS estimate of $\mathbf{M}$ can be uniquely determined by the Krylov controllability matrix of pair $\{\mathbf{U}'\mathbf{U}, \mathbf{U}'\mathbf{Y}\}$ and the coefficients of the characteristic polynomial of matrix $\mathbf{U}'\mathbf{U}$.

If $\mathbf{U}'\mathbf{U}$ is singular, the inverse of $\mathbf{U}'\mathbf{U}$ no longer exists. However, as proved by Di Ruscio (2000, [113]), one can use the following truncated Cayley-Hamilton series:

$$\mathbf{U}_{tchs} = [(\mathbf{U}'\mathbf{U})^{l_0-1} \quad (\mathbf{U}'\mathbf{U})^{l_0-2} \quad \cdots \quad \mathbf{U}'\mathbf{U} \quad \mathbf{I}_l] \begin{bmatrix} \psi_1^* \mathbf{I}_l \\ \vdots \\ \psi_{l_0}^* \mathbf{I}_l \end{bmatrix} \tag{6.19}$$

to obtain an optimal approximation of $(\mathbf{U}'\mathbf{U})^{-1}$, where $1 \leq l_0 < l$ is the rank of $\mathbf{U}'\mathbf{U}$, $\boldsymbol{\psi}^* = \begin{bmatrix} \psi_1^* & \psi_2^* & \cdots & \psi_{l_0}^* \end{bmatrix}' \in \Re^{l_0}$ is a parameter vector to be determined later, and the subscript "tchs" is the abbreviation of "truncated Cayley-Hamilton series". As a consequence, the estimate of $\mathbf{M}$ is

$$\begin{aligned} \hat{\mathbf{M}}' &= [(\mathbf{U}'\mathbf{U})^{l_0-1} \quad (\mathbf{U}'\mathbf{U})^{l_0-2} \cdots \quad \mathbf{I}_l] \begin{bmatrix} \psi_1^* \mathbf{I}_l \\ \vdots \\ \psi_{l_0}^* \mathbf{I}_l \end{bmatrix} \mathbf{U}'\mathbf{Y} \\ &= \left[(\mathbf{U}'\mathbf{U})^{l_0-1}\mathbf{U}'\mathbf{Y} \quad (\mathbf{U}'\mathbf{U})^{l_0-2}\mathbf{U}'\mathbf{Y} \quad \cdots \quad \mathbf{U}'\mathbf{Y}\right] \begin{bmatrix} \psi_1^* \mathbf{I}_m \\ \psi_2^* \mathbf{I}_m \\ \vdots \\ \psi_{l_0}^* \mathbf{I}_m \end{bmatrix} \\ &= \mathbf{K}_{l_0} \begin{bmatrix} \psi_1^* \mathbf{I}_m \\ \psi_2^* \mathbf{I}_m \\ \vdots \\ \psi_{l_0}^* \mathbf{I}_m \end{bmatrix} \end{aligned} \tag{6.20}$$

where

$$\mathbf{K}_{l_0} \equiv \left[(\mathbf{U}'\mathbf{U})^{l_0-1}\mathbf{U}'\mathbf{Y} \quad (\mathbf{U}'\mathbf{U})^{l_0-2}\mathbf{U}'\mathbf{Y} \quad \cdots \quad \mathbf{U}'\mathbf{Y}\right] \in \Re^{l \times (l_0 m)}$$

is the reduced Krylov matrix of pair $\{\mathbf{U}'\mathbf{U}, \mathbf{U}'\mathbf{Y}\}$ with order of l_0.

Eqn. 6.20 indicates that an estimate of $\mathbf{M}$ matrix can be achieved with the reduced Krylov controllability matrix $\mathbf{K}_{l_0}$ and the parameters $\boldsymbol{\psi}^*$. Since the former is composed of data matrices $\mathbf{U}$ and $\mathbf{Y}$ and is always available, all one must do to make such a pursuit possible is to estimate $\boldsymbol{\psi}^*$.

6.3.2 Optimal Estimation of Parameter Vector

According to the LS criterion, the parameter vector $\boldsymbol{\psi}^*$ must be determined such that the prediction error matrix

$$\mathbf{Y} - \mathbf{U}\hat{\mathbf{M}}' = \mathbf{Y} - \mathbf{U}\mathbf{K}_{l_0} \begin{bmatrix} \psi_1^*\mathbf{I}_m \\ \psi_2^*\mathbf{I}_m \\ \vdots \\ \psi_{l_0}^*\mathbf{I}_m \end{bmatrix} \tag{6.21}$$

will have a minimum squared Frobenius norm ([113]). We now introduce the column vector operator $Vec(\cdot)$. For example, $Vec(\mathbf{Y}) \in \Re^{Nm}$ is a column vector constructed from the data matrix $\mathbf{Y}$ by sequentially stacking each column of $\mathbf{Y}$ onto another. Consequently, minimizing the squared F-norm of error matrix $\mathbf{Y} - \mathbf{U}\hat{\mathbf{M}}'$ in Eqn. 6.21 is equivalent to minimizing the squared 2-norm of the stacked vector, $Vec(\mathbf{Y}) - \mathbf{U_L}\boldsymbol{\psi}^*$, where,

$$\mathbf{U}_L = \left[Vec\left(\mathbf{U}\left(\mathbf{U}'\mathbf{U}\right)^{l_0-1}\mathbf{U}'\mathbf{Y}\right) \;\; Vec\left(\mathbf{U}\left(\mathbf{U}'\mathbf{U}\right)^{l_0-2}\mathbf{U}'\mathbf{Y}\right) \;\; \cdots \;\; Vec\left(\mathbf{U}\mathbf{U}'\mathbf{Y}\right) \right]$$

is an $(Nm) \times l_0$ matrix. The LS solution to this minimization problem can be readily obtained:

$$\hat{\boldsymbol{\psi}}^* = \left(\mathbf{U}_L'\mathbf{U}_L\right)^{-1}\mathbf{U}_L' Vec(\mathbf{Y}) \tag{6.22}$$

6.3.3 Calculation of Latent Variables

We have shown in Section 6.2.2 that the weight vectors for $\mathbf{U}$ and $\mathbf{Y}$ can be calculated from Eqns. 6.12 and 6.13, provided that $(\mathbf{U}'\mathbf{U})^{-1}$ and $(\mathbf{Y}'\mathbf{Y})^{-1}$ exist.

When $\mathbf{U}'\mathbf{U}$ is singular (as explained earlier $\mathbf{Y}'\mathbf{Y}$ is assumed never to be singular), we use the truncated Cayley-Hamilton series, $\mathbf{U}_{tchs}$, to approximate $(\mathbf{U}'\mathbf{U})^{-1}$. Using such an approximation in Eqns. 6.12 and 6.13 leads to

$$\mathbf{U}_{tchs}\mathbf{U}'\mathbf{Y}\left(\mathbf{Y}'\mathbf{Y}\right)^{-1}\mathbf{Y}'\mathbf{U}\mathbf{p}_j = \lambda_j\mathbf{p}_j \tag{6.23}$$

and

$$(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{Y}\mathbf{w}_j = \lambda_j \mathbf{w}_j \tag{6.24}$$

Pre-multiplying Eqn. 6.23 by $\mathbf{U}$ gives

$$\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{Y}\,(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{U}\mathbf{p}_j = \lambda_j \mathbf{U}\mathbf{p}_j,$$

which can be simplified into

$$\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y\mathbf{U}\mathbf{p}_j = \lambda_j \mathbf{U}\mathbf{p}_j. \tag{6.25}$$

by denoting

$$\mathbf{P}_y \equiv \mathbf{Y}\,(\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'.$$

Similarly, pre-multiplying Eqn. 6.24 by $\mathbf{Y}$ generates

$$\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{h}_j = \lambda_j \mathbf{h}_j, \tag{6.26}$$

where $\mathbf{h}_j \equiv \mathbf{Y}\mathbf{w}_j$.

Eqns. 6.25 and 6.26 represent eigenproblems for non-symmetric matrices. Because only eigenvectors of a symmetric matrix have useful features, e.g. orthonormality, we must transform the non-symmetric eigenproblem into a symmetric one.

We begin the investigation with the calculation of $\mathbf{h}_j$ for $j \in [1, n_0]$. Noted that $\mathbf{P}_y$ is a projection matrix. Therefore, $\mathbf{P}_y^2 = \mathbf{P}_y$, and we can rewrite Eqn. 6.26 as

$$\mathbf{P}_y\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{h}_j = \lambda_j \mathbf{h}_j, \tag{6.27}$$

It is a well known fact that for two arbitrary matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ with compatible dimensions, $\mathbf{M}_1\mathbf{M}_2$ and $\mathbf{M}_2\mathbf{M}_1$ have identical eigenvalues. Thus, matrices $\mathbf{P}_y\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'$ and $\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y$ share eigenvalues.

Furthermore, since $\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y$ is symmetric, any existing standard algorithms for a symmetric eigenproblem can be utilized to calculate the latent vectors $\mathbf{h}_j$. For example, as shown in Golub (1973, [42]), if

$$\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y\mathbf{z}_j = \lambda_j \mathbf{z}_j \tag{6.28}$$

where $\mathbf{z}_j$ is the j^{th} eigenvector of the symmetric matrix, then

$$\mathbf{h}_j \equiv \mathbf{P}_y\mathbf{z}_j.$$

Finally, the calculation of another set of latent vector $\mathbf{t}_j$ for $\mathbf{U}$ is discussed. Because $\mathbf{U}_{tchs}$ is the best approximation of $(\mathbf{U}'\mathbf{U})^{-1}$, we can express the product of $\mathbf{U}_{tchs}$ and $\mathbf{U}'\mathbf{U}$ as follows:

$$\mathbf{U}_{tchs}\mathbf{U}'\mathbf{U} = \mathbf{I}_l + \boldsymbol{\varepsilon}(\mathbf{U})$$

where $\boldsymbol{\varepsilon}(\mathbf{U})$ represents the error caused by the approximation. Obviously, $\boldsymbol{\varepsilon}(\mathbf{U}) = \mathbf{0}$ when $\mathbf{U}'\mathbf{U}$ is not singular.

The incorporation of $\mathbf{I}_l = \mathbf{U}_{tchs}\mathbf{U}'\mathbf{U} - \boldsymbol{\varepsilon}(\mathbf{U})$ into Eqn. 6.25 tells us that

$$\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y\mathbf{U}\left(\mathbf{U}_{tchs}\mathbf{U}'\mathbf{U} - \boldsymbol{\varepsilon}(\mathbf{U})\right)\mathbf{p}_j = \lambda_j\mathbf{U}\mathbf{p}_j$$

which, because $\mathbf{t}_j \equiv \mathbf{U}\mathbf{p}_j$, can be rewritten as:

$$\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y\mathbf{U}\left(\mathbf{U}_{tchs}\mathbf{U}'\mathbf{t}_j - \boldsymbol{\varepsilon}(\mathbf{U})\mathbf{p_j}\right) = \lambda_j\mathbf{t}_j, \tag{6.29}$$

If we ignore the term $\boldsymbol{\varepsilon}(\mathbf{U})\mathbf{p_j}$ on the left hand side (LHS), Eqn. 6.29 can be further simplified as follows:

$$\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{t}_j = \lambda_j\mathbf{t}_j \tag{6.30}$$

In such a case, $\mathbf{t}_j$ is the j^{th} eigenvector of the symmetric matrix $\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'\mathbf{P}_y\mathbf{U}\mathbf{U}_{tchs}\mathbf{U}'$.

Having developed the algorithm to directly calculate latent vectors for $\mathbf{U}$ and $\mathbf{Y}$, the following remarks can be made.

Remark 6.3.1 Because $\{\mathbf{t}_1, \mathbf{t}_2 \cdots, \mathbf{t}_j\}$ are eigenvectors of a symmetric matrix, orthonormality among these latent vectors of $\mathbf{U}$ are ensured. The same conclusion can be applied to $\{\mathbf{h}_1, \mathbf{h}_2 \cdots, \mathbf{h}_j\}$, the latent vectors of $\mathbf{Y}$.

Remark 6.3.2 Due to the omission of $\boldsymbol{\varepsilon}(\mathbf{U})$, the orthogonality between $\mathbf{t}_i$ and $\mathbf{h}_j$ for $i \neq j$ is lost. This is slightly different from the conventional CVA, and is the price paid for handling the collinearity among $\mathbf{U}$. However, because the inverse of $\mathbf{U}'\mathbf{U}$ is optimally approximated by $\mathbf{U}_{tchs}$, $\mathbf{t}_i$ and $\mathbf{h}_i$ still have the highest correlation, while $\mathbf{t}_i$ and $\mathbf{h}_j$ have the least correlation when $i \neq j$.

Remark 6.3.3 The newly-developed CVA is reduced to conventional CVA if $\mathbf{U}'\mathbf{U}$ is not singular. When this is the case, $\mathbf{U}_{tchs} = (\mathbf{U}'\mathbf{U})^{-1}$. Hence, the latter is in fact a special case of the former.

Assuming that $\mathbf{Y}'\mathbf{Y}$ is non-singular, we can state the complete algorithm of the novel CVA developed in this section as follows:

Step 1 With N samples, construct data matrices $\mathbf{U}$ and $\mathbf{Y}$, and scale each variable in the data matrices into zero mean and/or unit variance.

Step 2 Calculate the condition number (CN) of matrix $\mathbf{U}'\mathbf{U}$. If the CN is large, e.g. 10^{10}, go directly to Step 4. Otherwise, go to Step 3.

Step 3 Perform conventional CVA on $\mathbf{U}$ and $\mathbf{Y}$. Stop.

Step 4 Calculate the rank, l_0, of $\mathbf{U}'\mathbf{U}$. Subsequently, form the matrix $\mathbf{U}_L$ and calculate $\hat{\boldsymbol{\psi}}^*$ according to Eqn. 6.22.

Step 5 Calculate $\mathbf{U}_{tchs}$ using Eqn. 6.19. Then calculate $\mathbf{P}_y$.

Step 6 Calculate $\{\mathbf{t}_j, \mathbf{h}_j\}$ using Eqns. 6.30 and 6.28, respectively, for $j \in [1, n_0]$ with $n_0 = \min(l, m)$.

Step 7 Using Eqns. 6.8 and 6.9, calculate the predicted value, $\hat{\mathbf{y}}(k)$, of $\mathbf{y}(k)$ and the prediction error, $\tilde{\mathbf{y}}(k)$, of $\mathbf{y}(k)$.

Step 8 If necessary, calculate $\tilde{\mathbf{h}}(k)$ using Eqn. 6.11. Note that $\boldsymbol{\Omega}_{n_0}$ is no longer a diagonal matrix due to the lose of orthogonality between $\mathbf{h}_i$ and $\mathbf{t}_j$ for $i, j \in [1, n_0$ and $i \neq j$.

6.4 Extension to Dynamic Processes

In this section, the newly developed CVA algorithm will be extended to dynamic processes. In accordance with Bauer and Ljung (2002, [11]), a linear time invariant multi-input multi-output (MIMO) process can be represented by the following state space equations in the discrete time domain:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\tilde{\mathbf{u}}(k) + \mathbf{E}\boldsymbol{\phi}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \boldsymbol{\phi}(k) \end{aligned} \tag{6.31}$$

The system under consideration is assumed to be stable, i.e. all eigenvalues of $\mathbf{A}$ lie inside the unit circle, and strictly minimum phase, i.e. the eigenvalues of $\mathbf{A}_k \equiv \mathbf{A} - \mathbf{EC}$ are also assumed to be located inside the unit circle. It is further assumed that the inputs $\tilde{\mathbf{u}}(k)$ are known, as they are outputs from controllers.

Having performed a series of algebraic manipulations on Eqn. 6.31, we can arrive at

$$\mathbf{y}_\mu(k) = [\mathbf{P}_\mu \mid \mathbf{H}_\mu] \begin{bmatrix} \tilde{\mathbf{u}}_{\mu-1}(k-\mu) \\ \mathbf{y}_{\mu-1}(k-\mu) \\ \tilde{\mathbf{u}}_{\mu-1}(k) \end{bmatrix} + \mathbf{G}_\mu \boldsymbol{\phi}_\mu(k). \tag{6.32}$$

$\mathbf{y}_\mu(k) \in \Re^{m_\mu}$, $\boldsymbol{\phi}_\mu(k) \in \Re^{n_\mu}$, $\{\tilde{\mathbf{u}}_{\mu-1}(k-\mu), \tilde{\mathbf{u}}_{\mu-1}(k)\} \in \Re^{l\mu}$, and $\mathbf{y}_{\mu-1}(k-\mu) \in \Re^{m\mu}$ are stacked vectors with $m_\mu \equiv m(\mu+1)$ and $n_\mu \equiv n(\mu+1)$. μ is a positive integer that can be calculated through methods developed by Bauer (2001, [10]). The definition of a stacked vector is

$$\mathbf{z}_\varrho(\tau) = \begin{bmatrix} \mathbf{z}(\tau) \\ \vdots \\ \mathbf{z}(\tau+\varrho) \end{bmatrix}$$

where $\mathbf{z}$ is equal to $\tilde{\mathbf{u}}$ or $\mathbf{y}$, τ to k or $k-\mu$, and ϱ to μ or $\mu-1$ respectively. Moreover, in Eqn. 6.32

$$\mathbf{P}_\mu = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^\mu \end{bmatrix} \begin{bmatrix} \mathbf{A}_k^{\mu-1}\mathbf{B} & \mathbf{A}_k^{\mu-2}\mathbf{B} & \cdots & \mathbf{B} \mid \mathbf{A}_k^{\mu-1}\mathbf{E} & \mathbf{A}_k^{\mu-2}\mathbf{E} & \cdots & \mathbf{E} \end{bmatrix} \in \Re^{m_\mu \times ((l+m)\mu)},$$

$$\mathbf{H}_\mu = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CB} & \mathbf{0} & \vdots \\ \vdots & \ddots & \\ \mathbf{CA}^{\mu-1}\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \in \Re^{m_\mu \times l\mu},$$

$$\mathbf{G}_\mu = \begin{bmatrix} \mathbf{I}_n & \cdots & & \mathbf{0} \\ \mathbf{CE} & \ddots & & \vdots \\ \vdots & \ddots & & \\ \mathbf{CA}^{\mu-1}\mathbf{E} & \cdots & \mathbf{CE} & \mathbf{I}_n \end{bmatrix} \in \Re^{m_\mu \times n_\mu}.$$

The detailed derivation of Eqn. 6.32 is provided in Appendix 6.A.

In Eqn. 6.32, because $\boldsymbol{\phi}_\mu(k)$ has been assumed to be a Gaussian distributed white noise sequence, it can be proved that $\mathbf{G}_\mu \boldsymbol{\phi}_\mu(k)$ follows a zero mean Gaussian

distribution with covariance $\mathbf{R}_{\mu,\phi} = \mathbf{G}_\mu (\mathbf{I}_{\mu+1} \otimes \mathbf{R}_\phi) \mathbf{G}'_\mu$, i.e. $\mathbf{G}_\mu \boldsymbol{\varepsilon}_\mu(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_e)$, where $\otimes$ is the Kronecker tensor product.

Furthermore, if we define the data matrices as follows:

$$\mathbf{Y}_d = \begin{bmatrix} \mathbf{y}'_\mu(\mu+1) \\ \mathbf{y}'_\mu(\mu+2) \\ \vdots \\ \mathbf{y}'_\mu(\mu+N) \end{bmatrix} \in \Re^{N \times m_\mu},$$

$$\mathbf{U}_d = \begin{bmatrix} \mathbf{u}'_{\mu-1}(1) & \mathbf{y}'_{\mu-1}(1) & \mathbf{u}'_{\mu-1}(\mu+1) \\ \mathbf{u}'_{\mu-1}(2) & \mathbf{y}'_{\mu-1}(2) & \mathbf{u}'_{\mu-1}(\mu+2) \\ & \vdots & \\ \mathbf{u}'_{\mu-1}(N) & \mathbf{y}'_{\mu-1}(N) & \mathbf{u}'_{\mu-1}(\mu+N) \end{bmatrix} \in \Re^{N \times ((2l+m)\mu)}$$

and denote $\mathbf{M}_d \equiv [\mathbf{P}_\mu \mid \mathbf{H}_\mu]$, then Eqn. 6.32 can be extended to

$$\mathbf{Y}_d = \mathbf{U}_d \mathbf{M}'_d + \mathbf{E}_d \tag{6.33}$$

where

$$\mathbf{E}_d = \begin{bmatrix} \boldsymbol{\phi}'_\mu(\mu+1)\mathbf{G}'_\mu \\ \boldsymbol{\phi}'_\mu(\mu+2)\mathbf{G}'_\mu \\ \vdots \\ \boldsymbol{\phi}'_\mu(\mu+N)\mathbf{G}'_\mu \end{bmatrix} \in \Re^{N \times n_\mu}.$$

In Eqn. 6.33, the subscript d in each matrix stands for *dynamic*.

Now, because Eqn. 6.33 has a format similar to that of Eqn. 6.3, the newly developed CVA algorithms can be directly applied to dynamic processes by simply replacing $\mathbf{U}$ and $\mathbf{Y}$ with $\mathbf{U}_d$ and $\mathbf{Y}_d$, respectively.

6.5 Numerical Examples and Discussions

In this section, numerical examples are provided to demonstrate the validity of the novel CVA algorithm and its advantages over PLS, PCR, and conventional CVA with respect to its predictive powers.

6.5.1 Example 1

In this example, we use real data from a pulp and paper mill. This set of data, which has 6 input variables ($l = 6$) 2 output variables ($m = 2$) and 38 samples (N=38), is presented in Appendix E of Di Ruscio (2000, [113]).

The two output variables, $y_1(k)$ and $y_2(k)$, are tensile and tear, both of which are important in describing paper quality. It would be highly beneficial to predict these variables from the inputs, $\{u_1(k),\ u_2(k),\ \cdots,\ u_6(k)\}$. Among the 6 inputs, $u_1(k)$ is the freeness, $\{u_2(k),\ u_3(k),\ u_4(k),\ u_5(k)\}$ represent fiber length distribution, and $u_6(k)$ is the shive content of the pulp. When the length distribution is precisely measured, we have a linear dependency, $u_2(k) + u_3(k) + u_4(k) + u_5(k) = 100$.

We construct two data matrices, $\mathbf{U} \in \Re^{38\times 6}$ and $\mathbf{Y} \in \Re^{38\times 2}$ from the inputs and outputs. In these matrices, each column is scaled to be mean-centered and of unit variance. Moreover, we form the reduced Krylov matrix as follows:

$$\mathbf{K}_5 = \left[\ (\mathbf{U}'\mathbf{U})^4\mathbf{U}'\mathbf{Y}\quad (\mathbf{U}'\mathbf{U})^3\mathbf{U}'\mathbf{Y}\quad (\mathbf{U}'\mathbf{U})^2\mathbf{U}'\mathbf{Y}\quad (\mathbf{U}'\mathbf{U})\mathbf{U}'\mathbf{Y}\quad \mathbf{U}'\mathbf{Y}\ \right] \in \Re^{6\times 10}$$

where, because matrix $\mathbf{U}'\mathbf{U}$ has rank 5, $l_0 = 5$. Because $\min(l_0, m) = 2$, we choose $n_0 = [1, 2]$. Two cases are presented.

Case 1

In this case, 30 data points from Sample 5 to Sample 34 in $\mathbf{U}$ and $\mathbf{Y}$ are used. First, we obtain the estimates of the parameter vector $\boldsymbol{\psi}^*$, which is:

$$\hat{\boldsymbol{\psi}}^* = \left[\ 15.0173\quad -77.2865\quad 158.3347\quad -118.7903\quad 26.7435\ \right]'$$

Then, we formulate $\mathbf{U}_{tchs}$ based on $\mathbf{U}'\mathbf{U}$ and the elements in $\hat{\boldsymbol{\psi}}^*$. The latent vectors $\{\mathbf{t}_1,\ \mathbf{t}_2\}$ and $\{\mathbf{h}_1,\ \mathbf{h}_2\}$ are calculated directly.

After obtaining these latent vectors, we calculate the predicted values of $\mathbf{Y}$ by

$$\hat{\mathbf{Y}} = \mathbf{T}_{n_0}\mathbf{T}'_{n_0}\mathbf{Y}$$

where $n_0 = 1$ or 2. The prediction error matrix is $\mathbf{Y} - \hat{\mathbf{Y}}$, whose squared F-norm is listed in Table 6.1. For purpose of comparison, we apply other latent variable-based approaches (such as CVA, PLS, and PCR) to the calculation of $\hat{\mathbf{Y}}$. The resulting $\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2$ corresponding to different approaches with different numbers of n_0 are also listed in Table 6.1. Therein, note that, due to the singularity in $\mathbf{U}'\mathbf{U}$, the conventional CVA is not applicable (N/A). In addition, although PLS and PCR are

Table 6.1: $\|\mathbf{Y}-\hat{\mathbf{Y}}\|_F^2$ calculated from 30 samples using various approaches in Example 1

n_0	Novel CVA	Conventional CVA	PLS	PCR
1	0.9893	N/A	1.0266	1.0422
2	0.9216	N/A	0.9923	1.0098
3	N/A	N/A	0.9343	0.9895
4	N/A	N/A	0.9236	0.9708
5	N/A	N/A	0.9216	0.9216

workable, they give larger prediction errors. Thus, the novel CVA provides the best prediction. For PLS and PCR, the number of latent variables can be up to the rank of $\mathbf{U}'\mathbf{U}$ (5 in this case). It is clear that, for predicting outputs, the use of 2 latent variables in the new CVA is equivalent to using 5 in PLS or PCR. Thus, the new CVA is much more efficient.

We have verified the orthogonality between any two latent vectors of $\{\mathbf{t}_1,\ \mathbf{t}_2\}$, and between those of $\{\mathbf{h}_1,\ \mathbf{h}_2\}$. The correlations between $\{\mathbf{h}_i,\ \mathbf{t}_j\}$ for $i,j = [1,2]$ are presented below:

$$\begin{bmatrix} Cor(\mathbf{h}_1,\mathbf{t}_1) & Cor(\mathbf{h}_1,\mathbf{t}_2) \\ Cor(\mathbf{h}_2,\mathbf{t}_1) & Cor(\mathbf{h}_2,\mathbf{t}_2) \end{bmatrix} = \begin{bmatrix} 0.91188573812187 & -0.00000015956836 \\ -0.00000036889936 & 0.39443818209650 \end{bmatrix}$$

Clearly, the orthogonality between $\mathbf{t}_1$ and $\mathbf{h}_2$ or $\mathbf{t}_2$ and $\mathbf{h}_1$ is almost preserved (their respective correlations are very small), although the inverse of $\mathbf{U}'\mathbf{U}$ is approximated in terms of the truncated Cayley-Hamilton series.

Case 2

The remaining 8 samples are used for validation. As before, we calculate $\|\mathbf{Y}-\hat{\mathbf{Y}}\|_F^2$ using the newly proposed CVA and other latent variable-based methods. The results are shown in Table 6.2. Once again, it can be concluded that the new CVA outperforms other approaches.

Table 6.2: $\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2$ calculated from the remaining 8 samples using various approaches in Example 1

n_0	Novel CVA	Conventional CVA	PLS	PCR
1	0.3360	N/A	0.3774	0.3782
2	0.3457	N/A	0.3964	0.4014
3	N/A	N/A	0.3438	0.3974
4	N/A	N/A	0.3465	0.3681
5	N/A	N/A	0.3457	0.3457

6.5.2 Example 2

In this example, data are generated from a simulated continuously stirred tank reactor (CSTR) process, which is depicted in Figure 6.1. In this process, the reaction is 1st

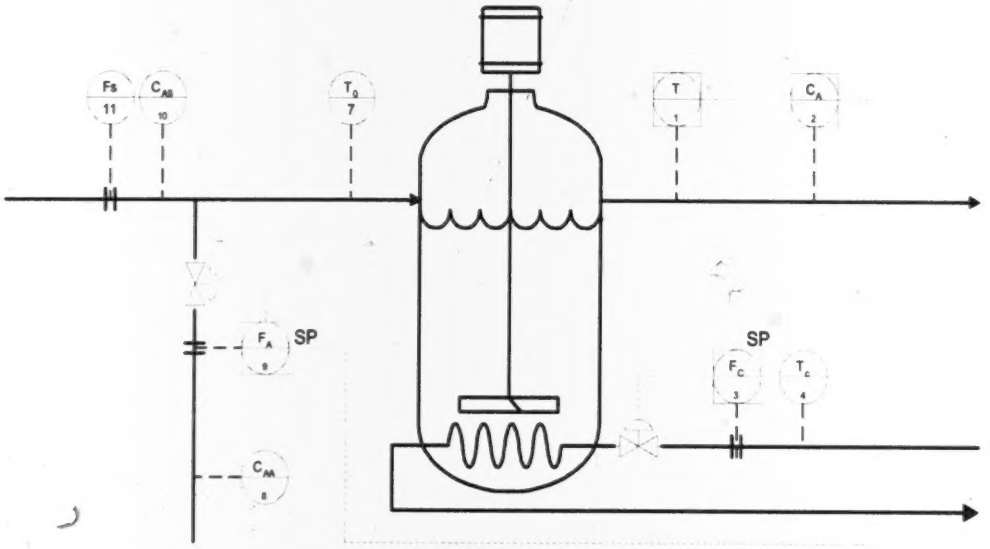

Figure 6.1: A simulated CSTR process

order and the reactor system involves heat transfer and heat of reaction. The process has (1) one feed stream, which is formed by merging solvent and the reactant, (2) one product stream, and (3) a coolant flowing to the coil. The flows of the feed stream and the coolant control the outlet concentration (C_A) and temperature (T). By assuming

that (1) the tank is well mixed; (2) physical properties are constant; and (3) the shaft work is negligible, the material and energy balances can be described by the following differential equations with nonlinear terms ([83]):

$$\begin{aligned} \frac{dC_A}{dt} &= \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-\frac{E}{RT}} C_A \\ \frac{dT}{dt} &= \frac{F}{V}(T_0 - T) + \frac{(-\Delta H_{rxn})}{\rho C_p} k_0 e^{-\frac{E}{RT}} C_A + \frac{aF_C^{b+1}(T - T_C)}{V\rho C_p \quad F_C + \frac{aF_C^b}{2\rho_c C_{pc}}} \end{aligned} \tag{6.34}$$

In Eqn. 6.34, F, C_{A0} and T_0 are the mass flow, concentration, and temperature of the feed stream respectively. F_C and T_C are the flow and temperature of the coolant, while

$$\{a, b, C_p, C_{pc}, E/R, k_0, -\Delta H_{rxn}, V, \rho, \rho_c\}$$

are process parameters. The values of these variables and parameters under normal operating conditions are given on pages 90 and 91 of Marlin (1995, [83]).

After linearizing Eqn. 6.34 around a steady operating point and omitting the deviation variables of F and T_0 for simplicity, one can represent the CSTR process in the continuous-time domain as follows:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{bmatrix} -7.5763 & -0.0935 \\ 854.9129 & 5.8153 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 & 0 \\ 0 & -12.1661 \end{bmatrix} \tilde{\mathbf{u}}(t) \\ \tilde{\mathbf{y}}(t) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(t) \end{aligned} \tag{6.35}$$

In Eqn. 6.35, the state variables $\mathbf{x}(t) \in \Re^2$ and the inputs $\tilde{\mathbf{u}}(t) \in \Re^2$ are the deviation variables of C_A, T, C_{A0}, and F_C, respectively, at the aforementioned steady operating point. In addition, $\tilde{\mathbf{y}}(t) \in \Re^2$ are the noise-free outputs.

Discretizing Eqn. 6.35 with a sampling rate of $T_s = 0.25min$ provides the counterpart of Eqn. 6.35 in the discrete-time domain:

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} -0.83143 & -0.01261 \\ 115.3000 & 0.97510 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.003698 & 0.02566 \\ 19.29000 & -3.72100 \end{bmatrix} \tilde{\mathbf{u}}(k) \\ \tilde{\mathbf{y}}(k) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) \end{aligned} \tag{6.36}$$

Moreover, by introducing the measurement noise, $\boldsymbol{\phi}(k)$, to the outputs, and

disturbances to the process, one can eventually arrive at

$$\begin{aligned}\mathbf{x}(k+1) &= \begin{bmatrix} -0.83143 & -0.01261 \\ 115.3000 & 0.97510 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} 0.003698 & 0.02566 \\ 19.29000 & -3.72100 \end{bmatrix}\tilde{\mathbf{u}}(k) \\ &\quad + \begin{bmatrix} 0.0000 & -0.0001 \\ -0.0001 & 0.0085 \end{bmatrix}\boldsymbol{\phi}(k) \qquad (6.37) \\ \mathbf{y}(k) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\mathbf{x}(k) + \boldsymbol{\phi}(k)\end{aligned}$$

where $\boldsymbol{\phi}(k)$ is a Gaussian-distributed white noise vector with covariance $\begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$.

We generate 100 points of training data by means of Eqn. 6.37, where we use the following MATLAB functions:

$$x_1 = idinput([N, 1], 'rbs', [0, 0.1], [-0.1, 0.1])$$
$$x_2 = idinput([N, 1], 'rbs', [0, 0.1], [-1, 1])$$

to simulate the two input sequences.

Assume that we have no *a prior* knowledge of the system order. Using the scheme developed by Bauer (2001, [10]), we determine it to be equal to 2. Subsequently, we construct the two data matrices $\mathbf{U}_d \in \Re^{96\times 12}$ and $\mathbf{Y}_d \in \Re^{96\times 6}$. We scale each column in $\mathbf{U}_d$ and $\mathbf{Y}_d$ into zero mean and unit variance.

The rank of $\mathbf{U}_d'\mathbf{U}_d$ is 12. We use the first 70 samples in $\mathbf{U}_d$ and $\mathbf{Y}_d$ for model identification, and the remainder for validation. As we did in Example 1, we calculate squared F-norm, $\|\mathbf{Y} - \hat{\mathbf{Y}}_d\|_F^2$, from the identification data with $n_0 \in [1, 6]$. The results are presented in Table 6.3. For comparison purpose, in the same table, we also include $\|\mathbf{Y} - \hat{\mathbf{Y}}_d\|_F^2$ calculated through the use of PCR and PLS, respectively, with $n_0 \in [1, 12]$, in the same table.

From the results provided in Tables 6.3 and 6.4, it can also be concluded that the newly proposed CVA performs better than PLS and PCR in a dynamic process.

6.6 Conclusion

A novel CVA algorithm has been proposed in this chapter. It preserves all the advantages of the conventional CVA, but has the additional advantage of being immune to linear dependency among input variables of a multivariate process.

Table 6.3: $\|\mathbf{Y} - \hat{\mathbf{Y}}_d\|_F^2$ F calculated from the first 70 data samples using various approaches in Example 2

n_0	Novel CVA	PLS	PCR
1	0.9993	1.2474	1.3096
2	0.6079	0.9336	0.9854
3	0.3882	0.6438	0.7866
4	0.3330	0.4692	0.6827
5	0.3128	0.3955	0.4814
6	0.3061	0.3535	0.4085
7	N/A	0.3411	0.3875
8	N/A	0.3313	0.3523
9	N/A	0.3237	0.3444
10	N/A	0.3173	0.3384
11	N/A	0.3090	0.3241
12	N/A	0.3061	0.3061

Table 6.4: $\|\mathbf{Y} - \hat{\mathbf{Y}}_d\|_F^2$ F calculated from the remaining data samples using various approaches in Example 2

n_0	Novel CVA	PLS	PCR
1	0.5577	0.6660	0.6784
2	0.3973	0.6534	0.7279
3	0.2990	0.5876	0.7774
4	0.2631	0.5174	0.6050
5	0.2455	0.3148	0.3973
6	0.2391	0.2645	0.3114
7	N/A	0.2444	0.3039
8	N/A	0.2552	0.2648
9	N/A	0.2447	0.2607
10	N/A	0.2422	0.2476
11	N/A	0.2405	0.2379
12	N/A	0.2391	0.2391

The significant contributions in this novel CVA are two-fold. Firstly, calculation of the latent vectors of the inputs and outputs has been transformed into a symmetric eigenproblem, in the format of truncated Cayley Hamilton series and a reduced Krylov controllability matrix. Secondly, the novel CVA has been extended to dynamic processes.

We have applied the newly proposed CVA algorithm to both simulated and real data with respect to predicting outputs. In addition, comparisons with other latent variable based schemes such as PLS, and PCR have been made in both static and dynamic cases. Results supporting the theoretical claim have been acquired.

6.A Derivation of Eqn. 6.32

By recursions, from Eqn. 6.31 it turns that

$$\begin{aligned}\mathbf{x}(k+i) &= \mathbf{A}^i\mathbf{x}(k)+\sum_{\tau=0}^{i-1}\mathbf{A}^\tau\left[\mathbf{B}\tilde{\mathbf{u}}(k+\tau)+\mathbf{E}\boldsymbol{\phi}(k+\tau)\right]\\ \mathbf{y}(k+i) &= \mathbf{C}\mathbf{A}^i\mathbf{x}(k)+\mathbf{C}\sum_{\tau=0}^{i-1}\mathbf{A}^\tau\left[\mathbf{B}\tilde{\mathbf{u}}(k+\tau)+\mathbf{E}\boldsymbol{\phi}(k+\tau)\right]+\boldsymbol{\phi}(k+i)\end{aligned} \quad (6.A.1)$$

where $i=[1,\mu]$. Therefore, stacking the preceding equation immediately gives

$$\mathbf{y}_\mu(k)=\boldsymbol{\Gamma}_\mu\mathbf{x}(k)+\mathbf{H}_\mu\mathbf{u}_{\mu-1}(k)+\mathbf{G}_\mu\boldsymbol{\phi}_\mu(k)$$

where $\boldsymbol{\Gamma}_\mu=\left[\mathbf{C}'\ (\mathbf{CA})'\ \cdots\ (\mathbf{CA}^\mu)'\right]'\in\Re^{m_\mu\times n}$.

On the other hand, substituting the second line for the first line of Eqn. 6.31 leads to

$$\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)+\mathbf{B}\tilde{\mathbf{u}}(k)+\mathbf{E}\mathbf{y}(k)$$

from which by recursion we can obtain

$$\begin{aligned}\mathbf{x}(k) &= \mathbf{A}^\mu\mathbf{x}(k-\mu)+\sum_{\tau=0}^{\mu-1}\mathbf{A}^\tau\left[\mathbf{B}\tilde{\mathbf{u}}(k-\mu+\tau)+\mathbf{E}\mathbf{y}(k-\mu+\tau)\right]\\ &\approx \sum_{\tau=0}^{\mu-1}\mathbf{A}^\tau\left[\mathbf{B}\tilde{\mathbf{u}}(k-\mu+\tau)+\mathbf{E}\mathbf{y}(k-\mu+\tau)\right] \quad (6.A.2)\\ &= \left[\mathbf{A}^{\mu-1}\mathbf{B}\ \ \mathbf{A}^{\mu-2}\mathbf{B}\ \cdots\ \mathbf{B}\mid\mathbf{A}^{\mu-1}\mathbf{E}\ \ \mathbf{A}^{\mu-2}\mathbf{E}\ \cdots\ \mathbf{E}\right]\begin{bmatrix}\tilde{\mathbf{u}}_{\mu-1}(k-\mu)\\ \mathbf{y}_{\mu-1}(k-\mu)\end{bmatrix}\end{aligned}$$

where $\mathbf{A}^{\mu}\mathbf{x}(k-\mu) \approx \mathbf{0}$ for a large value of μ, as pointed out by Gustafsson and Rao (2002, [46]).

Eventually, the combination of Eqns. 6.A.1 and 6.A.2 shows

$$\mathbf{y}_{\mu}(k) = [\mathbf{P}_{\mu}|\mathbf{H}_{\mu}] \begin{bmatrix} \tilde{\mathbf{u}}_{\mu-1}(k-\mu) \\ \mathbf{y}_{\mu-1}(k-\mu) \\ \tilde{\mathbf{u}}_{\mu-1}(k) \end{bmatrix} + \mathbf{G}_{\mu}\boldsymbol{\phi}_{\mu}(k)$$

which is identical to Eqn. 6.32.

7

Sensor and Actuator Fault Detection and Diagnosis Under Process Uncertainties[4]

7.1 Introduction

Early FDD methods assumed the availability of an accurate model of the monitored system. Because modelling errors, i.e. model-plant-mismatch (MPM), are always present in a complex system, in practice, such an assumption can be invalid. Moreover, disturbances are inevitable in most cases. Herein, MPM and process disturbances are referred as to *process uncertainties*, a term that will be used throughout this chapter. Process uncertainties can render most accurate model-based FDI schemes to be non-robust, making them unworkable in worst case.

[4]A version of this chapter has been published in *Control Engineering Practice* Vol. 15, pp. 587-599, 2005, by Han, Li and Shah.

Recently, robust FDD schemes that enable the detection and isolation of faults in the presence of process uncertainties have drawn increasing research attention. Basically, existing FDD schemes take process disturbances and MPM into consideration separately. Disturbances decoupling FDD methods include those developed by Frank in 1994 ([34]), and Patton and Chen in 1992 and 2000 ([100] [101]). FDD schemes with robustness against the modelling errors have been proposed by Lou *et al.* in 1986 ([74]), Frank and Ding in 1994 and 1997 ([32] [31]), Gertler and Kunwer in 1995 ([40]), Chen *et al.* in 1996 ([17]), Shen and Hsu in 1998 ([118]), Hamelin and Sauter in 2000 ([47]), Qin and Li in 2001 ([106]), and Li and Shah in 2002 ([70]). This has been accomplished in both time and frequency domains. However, unless some restrictive assumptions on the MPM are made ([136] [16]), very few FDD schemes have the capability of simultaneously working in the presence of both disturbances and MPM.

This chapter proposes an online and real time sensor and actuator FDD scheme that handles process disturbances and MPM simultaneously for a multivariate dynamic system. By extending the well-known Chow-Willsky approach ([19]), we generate a primary residual vector (PRV), which is a fault-accentuated signal for fault detection. To generate the PRV, one does not need a precise state space model of the system under consideration. A roughly estimated model is sufficient. To detect and isolate faults in the output sensors only, the PRV can be made totally insensitive to process uncertainties under some conditions. To detect and isolate faults in the actuators, the PRV can be made almost insensitive to process uncertainties.

7.2 Problem Formulation

7.2.1 System Description

Assume that the normal behavior of a multivariate dynamic process can be represented by the following discrete time linear state space model:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{E}\boldsymbol{\phi}(k) \\ \tilde{\mathbf{y}}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \tag{7.1}$$

where $\boldsymbol{\phi}(k) \in \Re^p$ represents the unmeasured deterministic process disturbance vector ([45]), which can be any unknown function of time. The process is assumed to be observable.

With the presence of sensor/actuator faults and instrument noise, the inputs to the process and the observed process outputs can be represented by:

$$\begin{aligned} \mathbf{u}(k) &= \tilde{\mathbf{u}}(k) + \mathbf{f}^u(k) \\ \mathbf{y}(k) &= \tilde{\mathbf{y}}(k) + \mathbf{f}^y(k) + \mathbf{o}(k) \end{aligned} \tag{7.2}$$

where $\mathbf{f}^u(k) \in \Re^l$ is the actuator fault; and $\mathbf{f}^y(k) \in \Re^m$ is the sensor fault. It is assumed that $\mathbf{o}_k$ is a Gaussian–distributed white noise vector with covariance matrix $\mathbf{R}_o$, and is independent of the initial state $\mathbf{x}_0$ and the disturbances $\boldsymbol{\phi}(k)$.

In the fault–free case, $\mathbf{f}^u(k)$ and $\mathbf{f}^y(k)$ are null vectors. In cases where some sensors/actuators are faulty, the corresponding elements in $\mathbf{f}^y(k)$ and $\mathbf{f}^u(k)$ will be non–zero, while the other elements remain zero. For example, to indicate that the first output sensor is faulty, the first element in $\mathbf{f}^y(k)$ is nonzero while other elements are zero. It is assumed that $\tilde{\mathbf{u}}(k)$ and $\mathbf{y}(k)$ are available, because they are controller outputs and the observed process outputs, respectively.

7.2.2 Process Uncertainties

In most cases, the true values of the system matrices $\{\mathbf{A},\ \mathbf{B}\}$ are never precisely known. However, an estimate $\{\mathbf{A}_\circ,\ \mathbf{B}_\circ\}$ of $\{\mathbf{A},\ \mathbf{B}\}$ can be available, and one has

$$\begin{aligned} \mathbf{A} &= \mathbf{A}_\circ + \delta\mathbf{A} \\ \mathbf{B} &= \mathbf{B}_\circ + \delta\mathbf{B} \end{aligned} \tag{7.3}$$

where $\{\delta\mathbf{A},\ \delta\mathbf{B}\}$ is the difference between $\{\mathbf{A},\ \mathbf{B}\}$ and $\{\mathbf{A}_\circ,\ \mathbf{B}_\circ\}$, representing the MPM. We assume that $\mathbf{C}$ is precisely known, i.e. $\mathbf{C} = \mathbf{C}_\circ$, because it is the sensor gain matrix. This is a widely accepted assumption in dealing with the problem of MPM ([136]).

The combination of Eqns. 7.1, 7.2 and 7.3 results in

$$\begin{aligned} \mathbf{x}(k+1) &= (\mathbf{A}_\circ + \delta\mathbf{A})\mathbf{x}(k) + (\mathbf{B}_\circ + \delta\mathbf{B})\mathbf{u}(k) + \mathbf{E}\boldsymbol{\phi}(k) \\ &= \mathbf{A}_\circ\mathbf{x}(k) + \mathbf{B}_\circ\mathbf{u}(k) + [\delta\mathbf{A}\ \ \delta\mathbf{B}] \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} + \mathbf{E}\boldsymbol{\phi}(k) \\ &= \mathbf{A}_\circ\mathbf{x}(k) + \mathbf{B}_\circ\tilde{\mathbf{u}}(k) + \mathbf{e}(k) + \mathbf{B}_\circ\mathbf{f}^u(k) \\ \mathbf{y}(k) &= \mathbf{C}_\circ\mathbf{x}(k) + \mathbf{o}(k) + \mathbf{f}^y(k) \end{aligned} \tag{7.4}$$

where $\mathbf{e}(k) \equiv \delta\mathbf{A}\mathbf{x}(k) + \delta\mathbf{B}\mathbf{u}(k) + \mathbf{E}\boldsymbol{\phi}(k) \in \Re^n$ is the process uncertainty vector accounting for the effects of the MPM and process disturbances. The following remarks can be made concerning Eqn. 7.4.

Remark 7.2.1 The gain matrix that links the process uncertainty vector $\mathbf{e}(k)$ and the state vector $\mathbf{x}(k+1)$ is an $n \times n$ identity matrix $\mathbf{I}_n$. With such a *constant* matrix, decoupling $\mathbf{e}(k)$ from the PRV is feasible, provided that certain conditions are met.

Remark 7.2.2 In the presence of actuator faults, $\mathbf{e}(k) = \delta\mathbf{A}\mathbf{x}(k) + \delta\mathbf{B}\tilde{\mathbf{u}}(k) + \delta\mathbf{B}\mathbf{f}^u(k) + \mathbf{E}\boldsymbol{\phi}(k)$. In this case, $\mathbf{e}(k) = \mathbf{e}^*(k) + \mathbf{e}^f(k)$, where $\mathbf{e}^*(k) = \delta\mathbf{A}\mathbf{x}(k) + \delta\mathbf{B}\tilde{\mathbf{u}}(k) + \mathbf{E}\boldsymbol{\phi}(k)$ is the fault-free portion, and $\mathbf{e}^f(k) = \delta\mathbf{B}\mathbf{f}^u(k)$ is the fault-related portion.

Remark 7.2.3 The assumption made on $\boldsymbol{\phi}(k)$ ([45])enables one to show that $\mathbf{e}^*(k)$ is a deterministic vector, if the initial state, i.e. $\mathbf{x}_o$, is not random. In addition, it is assumed that $\mathbf{e}^*(k)$ is bounded, e.g. $\|\mathbf{e}^*(k)\| \leq L_m$, where $\|\ \|$ stands for the L_2-norm. This means that process uncertainties only affect process dynamics to some extent. Note that $\mathbf{e}^*(k)$ consists of three terms, and $\|\mathbf{e}^*(k)\| = \|\delta\mathbf{A}\mathbf{x}(k)+\delta\mathbf{B}\tilde{\mathbf{u}}(k)+\mathbf{E}\boldsymbol{\phi}(k)\| \leq \|\delta\mathbf{A}\mathbf{x}(k)\|+\|\delta\mathbf{B}\tilde{\mathbf{u}}(k)\|+\|\mathbf{E}\boldsymbol{\phi}(k)\|$. If we further assume that each term has less energy acting on the system than the known input term $\|\mathbf{B}_o\tilde{\mathbf{u}}(k)\|$, then we can determine L_m to be equal to $\max_{k\in[1,\cdots]}\{3\|\mathbf{B}_o\tilde{\mathbf{u}}(k)\|\}$.

7.2.3 Problem of FDD in the Presence of Process Uncertainties

With Eqn. 7.4, the problem of FDD for the system of Eqn. 7.1 can be stated as follows:

1. From a set of training data, obtain an estimate $\{\mathbf{A}_o,\ \mathbf{B}_o,\ \mathbf{C}_o\}$ of the system matrices. This can be done by using any existing algorithm of SIM, e.g. the *N4SID* function in *Matlab*®;

2. In terms of the estimated system matrices, generate a PRV that is perfectly or almost perfectly independent of the process uncertainty vector $\mathbf{e}(k)$ for fault detection;

3. By manipulating the PRV algebraically, pinpoint the faulty sensors and/or actuators.

7.3 Robust Sensor FDD

This section is devoted to the detection and isolation of faults in the output sensors of the system under consideration. To achieve this goal, one must generate a PRV. The key to PRV generation is the derivation of a stacked equation.

7.3.1 PRV Generation

Starting from the time instant $k-s$, after performing a series of recursions on Eqn. 7.4, one arrives at

$$\mathbf{x}(k-s+i) = \mathbf{A}_\circ^i \mathbf{x}(k-s) + \sum_{\tau=k-s}^{k-s+i-1} \mathbf{A}_\circ^{k-s+i-1-\tau} \left[\mathbf{B}_\circ \tilde{\mathbf{u}}(\tau) + \mathbf{e}(\tau) + \mathbf{B}_\circ \mathbf{f}^u(\tau)\right]$$

and

$$\begin{aligned}\mathbf{y}(k-s+i) &= \mathbf{C}_\circ \mathbf{A}_\circ^i \mathbf{x}(k-s) + \mathbf{C}_\circ \sum_{\tau=k-s}^{k-s+i-1} \mathbf{A}_\circ^{k-s+i-1-\tau} \left[\mathbf{B}_\circ \tilde{\mathbf{u}}(\tau) + \mathbf{e}(\tau) + \mathbf{B}_\circ \mathbf{f}^u(\tau)\right] \\ &\quad + \mathbf{o}(k-s+i) + \mathbf{f}^y(k-s+i) \end{aligned} \tag{7.5}$$

where $i \in [1,\ s]$, s is the order of the *parity space* ([19]), and $\mathbf{x}(k-s)$ is the state vector at time instant $k-s$. In the following text, $s = n$ is selected.

By stacking Eqn. 7.5, it follows that

$$\begin{aligned}\mathbf{y}_s(k) - \mathbf{H}_s^\circ \tilde{\mathbf{u}}_{s-1}(k-1) &= \mathbf{\Gamma}_s^\circ \mathbf{x}(k-s) + \mathbf{f}_s^y(k) + \mathbf{H}_s^\circ \mathbf{f}_{s-1}^u(k-1) \\ &\quad + \mathbf{G}_s^\circ \mathbf{e}_{s-1}(k-1) + \mathbf{o}_s(k) \\ &= [\mathbf{\Gamma}_s^\circ \mid \mathbf{G}_s^\circ] \begin{bmatrix} \mathbf{x}(k-s) \\ \mathbf{e}_{s-1}(k-1) \end{bmatrix} + \mathbf{f}_s^y(k) \\ &\quad + \mathbf{H}_s^\circ \mathbf{f}_{s-1}^u(k-1) + \mathbf{o}_s(k) \end{aligned} \tag{7.6}$$

where, $\mathbf{y}_s(k) = \left[\begin{array}{ccc} \mathbf{y}'(k-s) & \cdots & \mathbf{y}'(k) \end{array}\right]' \in \Re^{m_s}$ is the stacked output vector,

$$\mathbf{\Gamma}_s^\circ = \left[\begin{array}{cccc} \mathbf{C}_\circ' & (\mathbf{C}_\circ \mathbf{A}_\circ)' & \cdots & (\mathbf{C}_\circ \mathbf{A}_\circ^s)' \end{array}\right]' \in \Re^{m_s \times n}$$

is the extended observability matrix, and

$$\mathbf{H}_s^\circ = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_\circ \mathbf{B}_\circ & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_\circ \mathbf{A}_\circ^{s-1} \mathbf{B}_\circ & \mathbf{C}_\circ \mathbf{A}_\circ^{s-2} \mathbf{B}_\circ & \cdots & \mathbf{C}_\circ \mathbf{B}_\circ \end{bmatrix} \in \Re^{m_s \times ls}$$

and

$$\mathbf{G}_s^\circ = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_\circ & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_\circ\mathbf{A}_\circ^{s-1} & \mathbf{C}_\circ\mathbf{A}_\circ^{s-2} & \cdots & \mathbf{C}_\circ \end{bmatrix} \in \Re^{m_s \times ns}$$

are two lower triangular block Toeplitz matrices with $m_s = m(s+1)$. In addition, $\tilde{\mathbf{u}}_{s-1}(k-1) \in \Re^{ls}$, $\mathbf{f}_s^y(k) \in \Re^{m_s}$, $\mathbf{f}_{s-1}^u(k-1) \in \Re^{ls}$, $\mathbf{e}_{s-1}(k-1) \in \Re^{ns}$ and $\mathbf{o}_s(k) \in \Re^{m_s}$ are also stacked vectors similar to $\mathbf{y}_s(k)$. Note that in Eqn. 7.6, the stacked uncertainty vector $\mathbf{e}_{s-1}(k-1)$ and the unknown state vector $\mathbf{x}(k-s)$ have gain matrices $\mathbf{G}_s^\circ$ and $\mathbf{\Gamma}_s^\circ$, respectively.

7.3.2 Sensor Fault Detection

Because in this section we only consider the detection and isolation of faults in the output sensors, we can simplify Eqn. 7.6 into

$$\mathbf{y}_s(k) - \mathbf{H}_s^\circ \tilde{\mathbf{u}}_{s-1}(k-1) = \mathbf{\Psi}_s^\circ \begin{bmatrix} \mathbf{x}(k-s) \\ \mathbf{e}_{s-1}(k-1) \end{bmatrix} + \mathbf{f}_s^y(k) + \mathbf{o}_s(k) \tag{7.7}$$

where $\mathbf{\Psi}_s^\circ \equiv [\mathbf{\Gamma}_s^\circ \mid \mathbf{G}_s^\circ] \in \Re^{m_s \times (n+ns)}$. We select a transformation matrix $\mathbf{W}_\circ$, which is located in the *left null space* of $\mathbf{\Psi}_s^\circ$, i.e. $\mathbf{W}_\circ \mathbf{\Psi}_s^\circ \equiv \mathbf{0}$, and has maximized covariance with the gain matrix $\mathbf{I}_{m_s}$ of $\mathbf{f}_s^y(k)$. According to the algorithm proposed by Li and Shah (2002, [70]), the solution to $\mathbf{W}_\circ$ is

$$\begin{aligned} \mathbf{W}_\circ' \;=\; & \text{eigenvectors associated with the non–zero} \\ & \text{eigenvalues of matrix } (\mathbf{\Psi}_s^\circ)^\perp \end{aligned} \tag{7.8}$$

where $(\mathbf{\Psi}_s^\circ)^\perp = \mathbf{I}_{m_s} - \mathbf{\Psi}_s^\circ(\mathbf{\Psi}_s^\circ)^\dagger$, and $(\mathbf{\Psi}_s^\circ)^\dagger$ stands for the Moore–Penrose pseudo inverse ([41]).

We assume that $\mathbf{\Psi}_s^\circ$ is of rank $ns+n$. This is the worst case with respect to uncertainty decoupling from the PRV. As a result, $(\mathbf{\Psi}_s^\circ)^\perp$ has at least $m_s - ns - n$ non-zero eigenvalues, and $\mathbf{W}_\circ'$ are the associated $m_s - ns - n$ eigenvectors.

Pre–multiplying both sides of Eqn. 7.7 by $\mathbf{W}_\circ$ leads to

$$\begin{aligned} \boldsymbol{\varepsilon}_s(k) &\equiv \mathbf{W}_\circ [\mathbf{I}_{m_s} \mid -\mathbf{H}_s^\circ] \begin{bmatrix} \mathbf{y}_s(k) \\ \mathbf{u}_{s-1}(k-1) \end{bmatrix} \\ &= \mathbf{W}_\circ \mathbf{f}_s^y(k) + \mathbf{W}_\circ \mathbf{o}_s(k) \in \Re^{m_s - ns - n} \end{aligned} \tag{7.9}$$

where $\mathbf{W}_o\Psi_s^o \equiv \mathbf{0}$ has been employed. As a result, both the unknown state vector $\mathbf{x}(k-s)$ and the process uncertainty vector $\mathbf{e}_{s-1}(k-1)$ have been removed from $\boldsymbol{\varepsilon}_s(k)$. We define $\boldsymbol{\varepsilon}_s(k)$ as a PRV for fault detection and make the following remarks:

Remark 7.3.1 *On the right hand side of Eqn. 7.9, the first line is the computational form, and* $\mathbf{W}_o[\mathbf{I}_{m_s} \mid -\mathbf{H}_s^o]$ *is referred to as the PRV model. In addition, the second line is the internal form showing how the output sensor faults affect the PRV.*

Remark 7.3.2 *Without a fault,* $\boldsymbol{\varepsilon}_s(k) = \boldsymbol{\varepsilon}_s^*(k) = \mathbf{W}_o\mathbf{o}_s(k)$, *which is a moving average (MA) of measurement noise* $\mathbf{o}(k)$. *From the distribution of* $\mathbf{o}(k)$, *it can be concluded that* $\boldsymbol{\varepsilon}_s^*(k)$ *is also a zero-mean Gaussian-distributed random vector ([58]) with covariance matrix* $\mathbf{R}_{s,\varepsilon} = \mathbf{W}_o\mathbf{R}_{s,o}\mathbf{W}_o^T$, *i.e.* $\boldsymbol{\varepsilon}_s^*(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{s,\varepsilon})$, *where* $\mathbf{R}_{s,o} = \mathbf{I}_{s+1} \otimes \mathbf{R}_o \in \Re^{m_s \times m_s}$ *is the covariance matrix of* $\mathbf{o}_s(k)$.

Remark 7.3.3 *With the occurrence of faults,*

$$\boldsymbol{\varepsilon}_s(k) = \boldsymbol{\varepsilon}_s^f(k) + \boldsymbol{\varepsilon}_s^*(k) \tag{7.10}$$

where $\boldsymbol{\varepsilon}_s^f(k) = \mathbf{W}_o\mathbf{f}_s^y(k)$ *is the fault-contribution term. In this case,* $\boldsymbol{\varepsilon}_s(k)$ *is a Gaussian-distributed random vector with mean* $\boldsymbol{\varepsilon}_s^f(k)$ *and covariance* $\mathbf{R}_{s,\varepsilon}$, *i.e.* $\boldsymbol{\varepsilon}_s(k) \sim \mathcal{N}(\boldsymbol{\varepsilon}_s^f(k), \mathbf{R}_{s,\varepsilon})$. *Suppose that* $\mathbf{W}_o\mathbf{f}_s^y(k) \neq \mathbf{0}$. *Consequently, fault detection can be conducted by simply checking if the mean of* $\boldsymbol{\varepsilon}_s(k)$ *has deviated from zero.*

One can define the following scalar squared weighted residual (SWR)

$$\eta_s(k) = \boldsymbol{\varepsilon}_s'(k)\mathbf{R}_{s,\varepsilon}^{-1}\boldsymbol{\varepsilon}_s(k) \tag{7.11}$$

as the fault detection index. In the fault-free case, $\boldsymbol{\varepsilon}_s(k) = \boldsymbol{\varepsilon}_s^*(k)$, correspondingly $\eta_s(k)$ follows a central chi-square distribution with $m_s - ns - n$ degrees of freedom ([58]), i.e. $\eta_s(k) \sim \chi^2(m_s - ns - n)$. However, if any sensor is faulty, $\eta_s(k)$ will no longer follow the central chi-square distribution ([8]). Therefore, fault detection can be carried out by comparing $\eta_s(k)$ with a predetermined threshold $\chi_\alpha^2(m_s - ns - n)$, where α is a selected level of significance, e.g. $\alpha = 5\%$. Whereas $\eta_s(k) < \chi_\alpha^2(m_s - ns - n)$ indicates that no fault occurs in sensors, $\eta_s(k) \geq \chi_\alpha^2(m_s - ns - n)$ implies that some sensors are faulty.

7.3.3 Sensor Fault Isolation

After faults have been detected, one must identify the faulty sensors. To achieve this goal, the PRV must be transformed into a set of SRVs, where one SRV is made insensitive to a subset of sensor faults but most sensitive to the other sensor faults.

Table 7.1: Incidence matrix to characterize the isolation logic of sensor faults

	$f_1(k)$	$f_2(k)$	$f_3(k)$	...	$f_m(k)$
$\mathbf{r}_s^1(k)$	0	1	1	$\cdots$	1
$\mathbf{r}_s^2(k)$	1	0	1	$\cdots$	1
$\mathbf{r}_s^3(k)$	1	1	0	$\cdots$	1
$\mathbf{r}_s^4(k)$	1	1	1	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{r}_s^m(k)$	1	1	1	$\cdots$	0

In the considered system, there are m sensors. Generally, to isolate faults in all m sensors, m SRVs must be generated. Without loss of generality, we consider the case of isolating a *single* sensor fault at each time. Correspondingly, we choose an incidence matrix given in Table 7.1 to characterize the SRVs' sensitivity and insensitivity to different faults. The selection of an incidence matrix is not unique. For detailed discussion, refer to Gertler and Singer ([36] [37]), and Li and Shah ([70]).

Denote

$$\mathbf{W}_{\circ,i} = [\mathbf{W}_\circ(:,i)\ \mathbf{W}_\circ(:,i+m)\ \cdots\ \mathbf{W}_\circ(:,i+ms)], \quad \forall\ i = [1,m]$$

where $\mathbf{W}_\circ(:,j)$ for $1 \le j \le m_s$ is the j^{th} column of $\mathbf{W}_\circ$. The stacked fault vector is $\mathbf{f}_s^y(k) = [(\mathbf{f}^y(k-s))'\ (\mathbf{f}^y(k-s+1))'\ \cdots\ (\mathbf{f}^y(k))']'$, and $\mathbf{W}_{\circ,i}$ contains the columns in $\mathbf{W}_\circ$ associated with the stacked i^{th} sensor fault $[0\ \cdots\ 0\ f_i^y(k-s)\ 0\ \cdots\ 0\ f_i^y(k-1)\ 0\ \cdots\ 0\ f_i^y(k)\ 0\ \cdots\ 0]'$ where $f_i^y(k-s)$ is the i^{th} element of $\mathbf{f}^y(k-s)$. In the incidence matrix given in Table 7.1, m SRVs are generated, where the i^{th} SRV: $\mathbf{r}_s^i(k)$ is insensitive to the afore-mentioned i^{th} stacked sensor fault, while it is most sensitive to all other faults. Further, a "0"/"1" corresponding to one SRV and one fault in the incidence matrix indicates that the SRV is designed to be insensitive/most sensitive to the fault. With such an incidence matrix, each fault can be isolated by observing different behavior of the SRVs. For example, if $\mathbf{r}_s^i(k)$ is not affected by a fault while all the other SRVs: $\mathbf{r}_s^2(k)$ to $\mathbf{r}_s^m(k)$ are, it can be concluded that a fault has occurred in the first sensor.

Mathematically, the i^{th} SRV is calculated by multiplying a transformation matrix

$\mathbf{W}_i$ on both sides of Eqn. 7.9:

$$\mathbf{r}_s^i(k) = \mathbf{W}_i\boldsymbol{\varepsilon}_s(k) = \mathbf{W}_i\mathbf{W}_\circ\mathbf{f}_s^y(k) + \mathbf{W}_i\mathbf{W}_\circ\mathbf{o}_s(k) \tag{7.12}$$

To ensure that $\mathbf{r}_s^i(k)$ is insensitive to the i^{th} sensor fault, $\mathbf{W}_i$ must be orthogonal to $\mathbf{W}_{\circ,i}$, i.e. $\mathbf{W}_i\mathbf{W}_{\circ,i} = \mathbf{0}$, $\forall\, i = [1, m]$.

In accordance with the algorithm given by Li and Shah (2002, [70]), one can calculate $\mathbf{W}_i$ as follows:

$$\begin{aligned}\mathbf{W}_i' \;=\; &\text{eigenvectors associated with the non-zero}\\ &\text{eigenvalues of matrix } \mathbf{W}_{\circ,i}^{\perp}\mathbf{W}_\circ\mathbf{W}_\circ'\end{aligned} \tag{7.13}$$

where $\mathbf{W}_{\circ,i}^{\perp} = \mathbf{I}_{m_s-ns-n} - \mathbf{W}_{\circ,i}\mathbf{W}_{\circ,i}^{\dagger}$. As a result, $\mathbf{r}_s^i(k)$ has $m_s - ns - n - (s+1)$ independent rows. Using $\mathbf{r}_s^i(k)$, one can similarly calculate the isolation indices $\eta_s^i(k) = (\mathbf{r}_s^i(k))'(\mathbf{R}_{s,\varepsilon}^i)^{-1}\mathbf{r}_s^i(k)$, $\forall\, i = [1, m]$, where $\mathbf{R}_{s,\varepsilon}^i = \mathbf{W}_i\mathbf{R}_{s,\varepsilon}\mathbf{W}_i'$ is the covariance matrix of $\mathbf{r}_s^i(k)$. Concerning the sensitivity or insensitivity to a fault, $\eta_s^i(k)$ is equivalent to $\mathbf{r}_s^i(k)$.

In accordance with the isolation logic similar to that defined in Table 7.1, if

$$\eta_s^i(k) \le \chi_\alpha^2(m_s - ns - n - s - 1),\ i \in [1, m];$$

and

$$\eta_s^j(k) \ge \chi_\alpha^2(m_s - ns - n - s - 1),\ \forall\, j \in [1, m] \cap \{j \neq i\}$$

then it can be concluded that the i^{th} output sensor is faulty.

7.3.4 Condition for Perfectly Decoupling the Uncertainty Vector

As shown in Eqn. 7.9, the PRV is $(m_s - ns - n)$-dimensional. To make the PRV perfectly uncorrelated with any process uncertainties, $m_s - ns - n = (m-n)(s+1) > 0$, i.e. $m - n > 0$, must be satisfied for fault detection. Furthermore, to leave enough degrees of freedom for the design of the SRVs for fault isolation, a more restrictive condition must also be met. For example, to isolate a single fault at each time in accordance with the isolation logic summarized in Table 7.1, $m_s - ns - n - (s+1) = (m - n - 1)(s+1) > 0$, i.e. $m - n - 1 > 0$, must be guaranteed.

However, $m - n - 1 > 0$ is not an unreasonable condition. For example, since most industrial processes have redundant and/or duplicate sensors for critical variables,

the number of sensors, i.e. m, is usually expected to be greater than the order of the model, i.e. n, in a chemical process. The model order, n, corresponds to the order of a reduced–complexity model that is able to capture the dominant dynamics of a process sufficiently well. In many industrial examples, model orders of $n \leq 2$ are sufficiently accurate and yet the process may have many more measurements, thus satisfying the inequality $m - n - 1 > 0$. Even when $m - n - 1 > 0$ cannot be satisfied, one can install redundant sensors on critical variables in order to increase m. This approach combines physical and analytical redundancy discussed in Chapter 2 to detect and also diagnose a fault.

7.4 Robust Actuator FDD

When there are faults in the actuators of a system under consideration, perfect decoupling of process uncertainties from the PRV is not achievable. This will be analyzed later. Nevertheless, decoupling the principal components of the uncertainties from the PRV is still feasible.

7.4.1 Difficulty in Completely Decoupling the Uncertainties from the PRV

For the sake of simplicity, we consider the actuator faults only, ignoring the sensor faults. Consequently, Eqn. 7.6 is reduced to

$$\mathbf{y}_s(k) - \mathbf{H}_s^{\circ}\tilde{\mathbf{u}}_{s-1}(k-1) = [\mathbf{\Gamma}_s^{\circ} \mid \mathbf{G}_s^{\circ}] \begin{bmatrix} \mathbf{x}(k-s) \\ \mathbf{e}_{s-1}(k-1) \end{bmatrix} + \mathbf{H}_s^{\circ}\mathbf{f}_{s-1}^{u}(k-1) + \mathbf{o}_s(k) \quad (7.14)$$

where the fault gain matrix is $\mathbf{H}_s^{\circ}$.

Note that $\mathbf{H}_s^{\circ} = \mathbf{G}_s^{\circ} \times \{\mathbf{I}_s \otimes \mathbf{B}_{\circ}\}$. If we select a matrix $\mathbf{W}_{\circ}$ that is orthogonal to $[\mathbf{\Gamma}_s^{\circ} \mid \mathbf{G}_s^{\circ}]$, i.e. $\mathbf{W}_{\circ}[\mathbf{\Gamma}_s^{\circ} \mid \mathbf{G}_s^{\circ}] = \mathbf{0}$ and pre-multiply Eqn. 7.14 by $\mathbf{W}_{\circ}$, we will have the resulting PRV:

$$\boldsymbol{\varepsilon}_s(k) = \mathbf{W}_{\circ}(\mathbf{y}_s(k) - \mathbf{H}_s^{\circ}\tilde{\mathbf{u}}_{s-1}(k-1)) = \mathbf{W}_{\circ}\mathbf{o}_s(k) \quad (7.15)$$

In the preceding equation, in addition to $\mathbf{x}(k-s)$ and $\mathbf{e}_{s-1}(k-1)$, the fault-contribution term $\mathbf{H}_s^{\circ}\mathbf{f}_{s-1}^{u}(k-1)$ is also removed, indicating that the PRV is insensitive to both the faults and process uncertainties. Therefore, no fault is detectable.

7.4.2 PRV Generation

To detect and isolate the actuator faults, one must compromise the design of the PRV, i.e. make the PRV insensitive to the principal components of the process uncertainties but sensitive to the faults as much as possible.

Performing the singular value decomposition (SVD) ([41]) on matrix $\mathbf{G}_s^\circ$ results in

$$\mathbf{G}_s^\circ = \mathbf{U}_G \mathbf{S}_G \mathbf{V}_G' \tag{7.16}$$

where $\mathbf{S}_G \in \Re^{ns \times ns}$ is a diagonal matrix with singular values in decreasing order, and $\mathbf{U}_G \in \Re^{m_s \times ns}$ and $\mathbf{V}_G \in \Re^{ns \times ns}$ contain the left and right singular vectors, respectively.

Eqn. 7.16 can be further separated into two parts, e.g.,

$$\mathbf{G}_s^\circ = \mathbf{U}_G \mathbf{S}_G \mathbf{V}_G' = \mathbf{U}_{G,1} \mathbf{S}_{G,1} \mathbf{V}_{G,1}' + \mathbf{U}_{G,2} \mathbf{S}_{G,2} \mathbf{V}_{G,2}' \tag{7.17}$$

where $\mathbf{S}_{G,1} \in \Re^{n_0 \times n_0}$ is the main submatrix of $\mathbf{S}_G$ containing n_0 principal singular values with $1 \leq n_0 < ns$, and $\{\mathbf{U}_{G,1},\ \mathbf{V}_{G,1}\}$ are the associated principal left and right singular vectors in $\{\mathbf{U}_G,\ \mathbf{V}_G\}$. In addition, $\mathbf{S}_{G,2} \in \Re^{(ns-n_0)\times(ns-n_0)}$ is the remaining submatrix of $\mathbf{S}_G$, and $\{\mathbf{U}_{G,2},\ \mathbf{V}_{G,2}\}$ are the associated remaining columns in $\{\mathbf{U}_G,\ \mathbf{V}_G\}$. The choice of n_0 will be discussed later.

Substituting Eqn. 7.17 for Eqn. 7.14 gives

$$\begin{aligned} \mathbf{y}_s(k) - \mathbf{H}_s^\circ \tilde{\mathbf{u}}_s(k) &= \mathbf{\Gamma}_s^\circ \mathbf{x}(k-s) + \mathbf{U}_{G,1}\mathbf{S}_{G,1}\mathbf{V}_{G,1}'\mathbf{e}_{s-1}(k-1) + \\ &\quad \mathbf{U}_{G,2}\mathbf{S}_{G,2}\mathbf{V}_{G,2}'\mathbf{e}_{s-1}(k-1) + \mathbf{H}_s^\circ \mathbf{f}_{s-1}^u(k-1) + \mathbf{o}_s(k) \end{aligned} \tag{7.18}$$

Denote $\mathbf{\Psi}_{s,1}^\circ = [\mathbf{\Gamma}_s^\circ \mid \mathbf{U}_{G,1}] \in \Re^{m_s \times (n+n_0)}$. Following the method shown in Section 7.3, we design a $\mathbf{W}_\circ$ such that it is orthogonal to $\mathbf{\Psi}_{s,1}^\circ$, i.e. $\mathbf{W}_\circ \mathbf{\Psi}_{s,1}^\circ \equiv 0$, while having a maximized covariance with $\mathbf{H}_s^\circ$. Mathematically,

$$\begin{aligned} \mathbf{W}_\circ' &= \text{eigenvectors associated with the non-zero} \\ &\quad \text{eigenvalues of matrix } (\mathbf{\Psi}_{s,1}^\circ)^\perp \mathbf{H}_s^\circ (\mathbf{H}_s^\circ)' \end{aligned} \tag{7.19}$$

where similarly $(\mathbf{\Psi}_{s,1}^\circ)^\perp = \mathbf{I}_{m_s} - \mathbf{\Psi}_{s,1}^\circ (\mathbf{\Psi}_{s,1}^\circ)^\dagger$. As a consequence, the PRV is

$$\begin{aligned} \boldsymbol{\varepsilon}_s(k) &= \mathbf{W}_\circ \mathbf{H}_s^\circ \mathbf{f}_{s-1}^u(k-1) + \mathbf{W}_\circ \mathbf{U}_{G,2}\mathbf{S}_{G,2}\mathbf{V}_{G,2}'\mathbf{e}_{s-1}(k-1) \\ &\quad + \mathbf{W}_\circ \mathbf{o}_s(k) \in \Re^{m_s - n_0 - n} \end{aligned} \tag{7.20}$$

where from Remark 7.2.2, $\mathbf{e}_{s-1}(k-1) = \mathbf{e}_{s-1}^*(k-1) + (\mathbf{I}_s \otimes \delta\mathbf{B})\mathbf{f}_{s-1}^u(k-1)$, and $\mathbf{e}_{s-1}^*(k-1)$ is stacked from $\mathbf{e}^*(k)$.

The PRV can be further decomposed as

$$\boldsymbol{\varepsilon}_s(k) = \boldsymbol{\varepsilon}_s^*(k) + \boldsymbol{\varepsilon}_s^f(k) \tag{7.21}$$

where $\boldsymbol{\varepsilon}_s^f(k) = [\mathbf{W}_o\mathbf{H}_s^\circ + \mathbf{M}(\mathbf{I}_s \otimes \delta\mathbf{B})]\,\mathbf{f}_{s-1}^u(k-1)$ is the fault–contribution term, and $\boldsymbol{\varepsilon}_s^*(k) = \mathbf{W}_o\mathbf{o}_s(k) + \mathbf{M}\mathbf{e}_{s-1}^*(k-1)$ is the fault–free term with $\mathbf{M} = \mathbf{W}_o\mathbf{U}_{G,2}\mathbf{S}_{G,2}\mathbf{V}'_{G,2}$.

It was mentioned in Remark 7.2.3 that $\mathbf{e}^*(k)$ is deterministic. As a result, the stacked vector $\mathbf{e}_{s-1}^*(k-1)$ is also deterministic. Due to the existence of the process uncertainty-related term $\mathbf{M}\mathbf{e}_{s-1}^*(k-1)$, the fault-free term $\boldsymbol{\varepsilon}_s^*(k)$ in the PRV is no longer zero mean. Instead, its mean is $E\{\boldsymbol{\varepsilon}_s^*(k)\} = \mathbf{M}\mathbf{e}_{s-1}^*(k-1)$.

In the fault-free case, the fault detection index $\eta_s(k) = \boldsymbol{\varepsilon}_s'(k)\mathbf{R}_{s,\varepsilon}^{-1}\boldsymbol{\varepsilon}_s(k)$ is reduced to $\eta_s^*(k) = (\boldsymbol{\varepsilon}_s^*(k))'\mathbf{R}_{s,\varepsilon}^{-1}\boldsymbol{\varepsilon}_s^*(k)$, which follows a noncentral chi-square distribution with $m_s - n - n_0$ degrees of freedom and noncentrality parameter $\|\mathbf{M}\mathbf{e}_{s-1}^*(k-1)\|^2$, i.e. $\eta_s^*(k) \sim \chi^2(m_s - n - n_0, \|\mathbf{M}\mathbf{e}_{s-1}^*(k-1)\|^2)$ ([104]).

Denote the maximum eigenvalue of $\mathbf{M}'\mathbf{M}$ by λ_{max}^M. From the Courant-Fischer Minimax Theorem ([41]), it can be inferred that $\|\mathbf{M}\mathbf{e}_{s-1}^*(k-1)\|^2 \leq \|\mathbf{e}_{s-1}^*(k-1)\|^2\lambda_{max}^M$. Furthermore, it follows from $\|\mathbf{e}^*(k)\| \leq L_m$ that $\|\mathbf{e}_{s-1}^*(k-1)\|^2 \leq s^2L_m^2$, indicating that the upper limit of the noncentrality parameter is $s^2L_m^2\lambda_{max}^M$. Therefore, with a selected level of significance α, one can choose $\chi_\alpha^2(m_s - n - n_0, s^2L_m^2\lambda_{max}^M)$ as the threshold for $\eta_s(k)$. Then fault detection can be similarly carried out by comparing $\eta_s(k)$ with the threshold.

Eventually, to isolate faults, one can transform the PRV into a set of SRVs, as was done in Section 7.3.3. One can also use the isolation logic in Table 7.1 to determine the sensitivity and insensitivity of the SRVs in relation to a fault. Moreover, using the SRVs, one can construct the fault isolation indices $\eta_s^i(k)$, $i = 1, \cdots, l$. After determining the thresholds for each $\eta_s^i(k)$, fault isolation can be performed in a manner similar to that in Section 7.3.3.

7.4.3 Conditions for Decoupling the Principal Components of Uncertainties from the PRV and Fault Detectability

Selecting Number of PCs

Denote $Rank\{(\boldsymbol{\Psi}_{s,1}^\circ)^\perp\mathbf{H}_s^\circ(\mathbf{H}_s^\circ)'\} = q$. From Eqns. 7.19 and 7.20, we can conclude that the dimension of the generated PRV is q, i.e. the PRV has q independent elements. To get a non-trivial solution to the PRV, at least, $q \geq 1$ must be guaranteed. Further, if one uses the isolation logic given in Table 7.1 to design a set of SRVs, each SRV will

have $q-s$ independent elements. Therefore, to ensure the isolation of a single actuator fault at one time, $q-s \geq 1$ must always hold. Finally, under the constraint $q-s \geq 1$, a larger n_0 is preferred. As a consequence, more components in the uncertainty vector $\mathbf{e}^*_{s-1}(k-1)$ will be removed from the PRV.

Fault Detectability with Process Uncertainties

It follows from Eqn. 7.21 that the mean of $\boldsymbol{\varepsilon}_s(k)$ in the presence of actuator faults is

$$E\{\boldsymbol{\varepsilon}_s(k)\} = \{\mathbf{W}_\circ\mathbf{H}^\circ_s + \mathbf{M}(\mathbf{I}_s \otimes \delta\mathbf{B})\}E\{\mathbf{f}^u_{s-1}(k-1)\} + \mathbf{M}\mathbf{e}^*_{s-1}(k-1) \qquad (7.22)$$

In this case, the fault detection index $\eta_s(k)$ is a non-central chi-square distributed random variable with non-centrality parameter $\|\mathbf{M}\mathbf{e}^*_{s-1}(k-1) + [\mathbf{M}(\mathbf{I}_s \otimes \delta\mathbf{B}) + \mathbf{W}_\circ\mathbf{H}^\circ_s]E\{\mathbf{f}^u_{s-1}(k-1)\}\|^2$. To ensure the detectability of actuator faults, $\|\mathbf{M}\mathbf{e}^*_{s-1}(k-1) + [\mathbf{M}(\mathbf{I}_s \otimes \delta\mathbf{B}) + \mathbf{W}_\circ\mathbf{H}^\circ_s]E\{\mathbf{f}^u_{s-1}(k-1)\}\|^2 \geq s^2L_m^2\lambda^M_{max}$ must hold true.

To simplify the above detectability condition, we denote

$$FDR(k) = \frac{[\mathbf{M}(\mathbf{I}_s \otimes \delta\mathbf{B}) + \mathbf{W}_\circ\mathbf{H}^\circ_s]E\{\mathbf{f}^u_{s-1}(k-1)\}}{\mathbf{M}\mathbf{e}^*_{s-1}(k-1)}$$

$$BDR(k) = \frac{sL_m\sqrt{\lambda^M_{max}}}{\mathbf{M}\mathbf{e}^*_{s-1}(k-1)}$$

where $FDR(k)$ and $BDR(k)$ represent fault-to-disturbance ratio and boundary-to-disturbance ratio at time instant k, respectively. Thus, the detectability condition can be rewritten as $\|\mathbf{l}_{ns} + FDR\|^2 \geq \|BDR\|^2$, where $\mathbf{l}_{ns}$ is a one vector with dimension ns. Therefore, if the FDD system has been designed, i.e. the matrices $\mathbf{M}$ and $\mathbf{W}_\circ$ are fixed, the detectability condition of actuator faults depends on the relationship between the fault-to-disturbance ratio and the boundary-to-disturbance ratio.

7.5 Numerical Example and Experimental Case Study

In this section, a numerical example and a real experimental case study are provided to demonstrate the validity of the proposed robust FDD scheme. We begin with the numerical example first, which includes FDD of sensors and actuators faults.

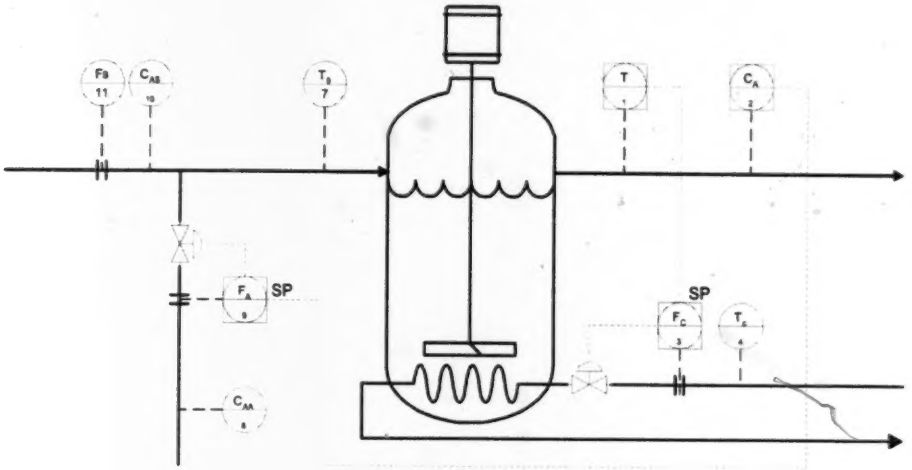

Figure 7.1: Process schematic of the simulated nonisothermal CSTR system

7.5.1 Numerical Example

The Simulated CSTR Process

Consider a simulated nonisothermal continuous stirred tank reactor (CSTR) process ([83]) as depicted in Figure 7.1. The process has (1) one feed stream, which is merged from the solvent and the reactant, (2) one product stream, and (3) a coolant flowing through the coil. The flows of the reactant (F_A) and the coolant (F_C) are used to control the residual concentration (C_A) and the outlet temperature (T), respectively. In addition, the reactant concentration in the solvent feed (C_{AS}) and the inlet temperature (T_o) are simulated as unmeasured disturbances.

The simulation parameters and initial conditions are selected to be the same as in Yoon and MacGregor (2001, [134]). Note that variables in the CSTR process, e.g. T and C_A, are functions of time. We omit the argument of the process variables for the sake of simplicity.

After being linearized around a steady operating point, the simulated CSTR process is represented by a second order continuous-time state space model with two inputs, four outputs and two unmeasured disturbances. In the model, the state variables are $\mathbf{x}(t) = [C_A \;\; T]'$, the inputs are $\mathbf{u}(t) = [F_C \;\; F_A]'$, and the disturbances are $\boldsymbol{\phi}(t) = [C_{AS} \;\; T_o]'$.

The temperature T and the residual concentration C_A are controlled by two

proportional controllers with unit gains. After discretizing the continuous-time system model using a sampling period $T_s = 0.5$ minute, the following discrete-time state space model is obtained:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\tilde{\mathbf{u}}(k) + \mathbf{E}\boldsymbol{\phi}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{o}(k) \end{aligned} \tag{7.23}$$

where,

$$\mathbf{A} = \begin{bmatrix} 0.2828 & -0.0005939 \\ 1.258 & 0.04251 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.0002216 & 0.5398 \\ -0.1042 & 1.576 \end{bmatrix},$$
$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}^T, \quad \mathbf{E} = \begin{bmatrix} 0.2844 & -0.00032506 \\ 0.68872 & 0.15287 \end{bmatrix}.$$

In addition, $\mathbf{x}(k)$, $\tilde{\mathbf{u}}(k)$ and $\boldsymbol{\phi}(k)$ are sampled values of $\mathbf{x}(t)$, $\mathbf{u}(t)$ and $\boldsymbol{\phi}(t)$ at $t = kT_s$, respectively, and $\mathbf{o}(k)$ is a Gaussian distributed white noise vector with covariance $diag([2.5 \times 10^{-5},\ \ 4 \times 10^{-4},\ \ 2.5 \times 10^{-5},\ \ 4 \times 10^{-4}])$. Note that in the preceding equation, there are four outputs (two sensors for C_A and T, and two additional sensors are introduced to satisfy the condition for perfect uncertainty decoupling).

We collected 400 samples of training data to identify the system matrices by means of the *N4SID* function in *Matlab*®, and obtain one estimate of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as follows:

$$\mathbf{A}_\circ = \begin{bmatrix} 0.35548 & -0.025014 \\ 1.3213 & 0.020207 \end{bmatrix}, \quad \mathbf{B}_\circ = \begin{bmatrix} -0.0084123 & 0.53482 \\ -0.11132 & 1.5683 \end{bmatrix}, \quad \mathbf{C}_\circ = \mathbf{C}$$

Notice that $\mathbf{A}_\circ$ and $\mathbf{B}_\circ$ are apparently biased from their true values. For example, the step response of the real system (solid line) and the estimated system (dashed line) shown in Figure 7.2 are different due to the MPM. As will be seen later, the presence of MPM may make the conventional Chow–Willsky method unworkable.

Sensor FDD for the CSTR

Using $\mathbf{A}_\circ$, $\mathbf{B}_\circ$ and $\mathbf{C}_\circ$, we construct $\boldsymbol{\Psi}_s^\circ \in \Re^{12\times 6}$ and $\mathbf{H}_s^\circ \in \Re^{12\times 4}$. Subsequently, we calculate $\mathbf{W}_\circ \in \Re^{6\times 12}$ according to Eqn. 7.8, obtaining the PRV model $\mathbf{W}_\circ[\mathbf{I}_{m_s} \mid -\mathbf{H}_s^\circ] \in \Re^{6\times 16}$. Further, based on the isolation logic shown in Table 7.1 ($m = 4$), we calculate four transformation matrices: $\mathbf{W}_i \in \Re^{3\times 6}$, $\forall\ i = [1, 4]$ to generate four SRVs, respectively, for fault isolation.

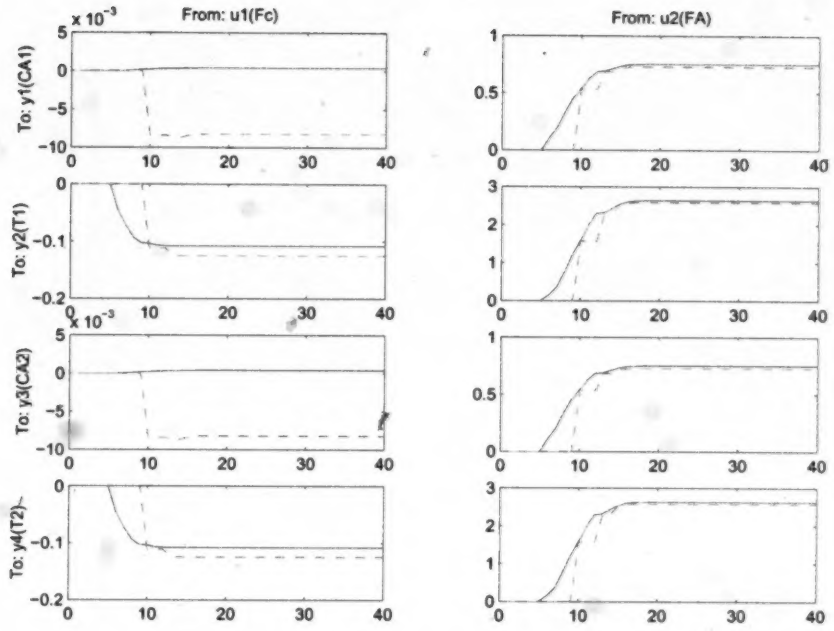

Figure 7.2: Step responses from the true model and its identified model of the simulated CSTR process (Solid line: Real system step response; Dashed line: Model step response)

Using the calculated PRV model and the same training data, we produce a sequence of PRV, from which the covariance matrices $\mathbf{R}_{s,\varepsilon} \in \Re^{6\times 6}$ and $\mathbf{R}^i_{s,\varepsilon} = \mathbf{W}_i\mathbf{R}_{s,\varepsilon}\mathbf{W}'_i \in \Re^{3\times 3}$, $\forall\ i = [1,4]$, are estimated. Because the PRV is 6-dimensional, the fault detection index $\eta_s(k) = \boldsymbol{\varepsilon}'_s(k)\mathbf{R}^{-1}_{s,\varepsilon}\boldsymbol{\varepsilon}_s(k)$ in the fault-free case is a chi-square random variable with 6 degrees of freedom, i.e. $\eta_s(k) \sim \chi^2(6)$. Given a specific level of significance, e.g. $\alpha = 0.01$, the confidence limit for $\eta_s(k)$ is $\chi^2_{0.01}(6) = 16.812$. Moreover, we compute four fault isolation indices $\eta^i_s(k)$, $\forall\ i = [1,4]$. With the same α, their confidence limits are $\chi^2_{0.01}(3) = 11.341$.

Four types of faults ([105]) – bias, drift, complete failure, and precision degradation – are simulated in this example. Due to the lack of space, only results for detection and isolation of a bias type fault is presented in this paper.

A bias $f^i(k) = 0.05$ is introduced to one of the four sensors from 500 to 700 sample instants. From the test data, $\eta_s(k)$ for fault detection and $\eta^i_s(k)$, $\forall\ i = [1,4]$, for fault isolation are calculated. For better visualization, $\eta_s(k)$ and $\eta^i_s(k)$, $\forall\ i = [1,4]$, have been scaled by their respective confidence limits, i.e. $\bar{\eta}_s(k) = \dfrac{\eta_s(k)}{\chi^2_{0.01}(6)}$ and $\bar{\eta}^i_s(k) = \dfrac{\eta^i_s(k)}{\chi^2_{0.01}(3)}$. As a result, $\bar{\eta}_s(k)$ and $\bar{\eta}^i_s(k)$ have a common unit confidence limit.

The FDD results are presented in Figure 7.3, where FD stands for $\bar{\eta}_s(k)$ and FI_i, $\forall\ i = [1,4]$ for $\bar{\eta}^i_s(k)$. As can be seen clearly in the figure, a fault is detected at $k = 500$. In addition, the first fault isolation index $\bar{\eta}^1_s(k)$ is within the confidence limit all the time, while the other isolation indices $\bar{\eta}^i_s(k)$, $\forall\ i = [2,4]$, are all beyond the limit after the occurrence of the fault. This produces an incidence code [0 1 1 1], from which one can correctly infer that the first output sensor is faulty.

In order to demonstrate the advantage of the proposed robust FDD method, a comparison with the original Chow–Willsky scheme ([19]) is made. The Chow–Willsky approach-based FDD results using the same test data and the same estimated state space model are given in Figure 7.4. In this figure, due to the effect of process uncertainties, the fault detection index and some of the fault isolation indices are outside their respective thresholds even after the fault has disappeared. For example, the first fault isolation index $\bar{\eta}^1_s(k)$ is beyond the limit, producing an incidence code [1 1 1 1] that violates the pre–determined incidence code [0 1 1 1]; hence the fault is not isolated correctly.

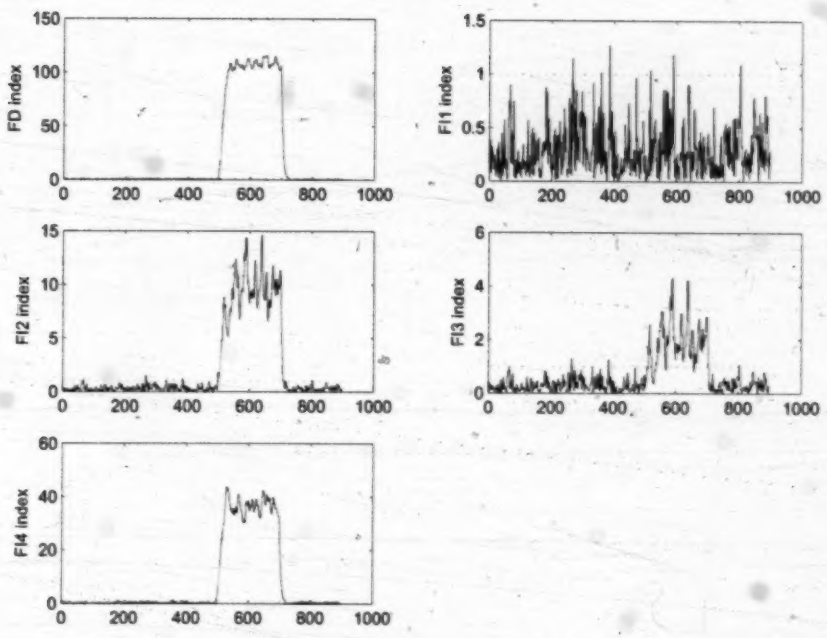

Figure 7.3: Detection and isolation of a bias fault in the 1^{st} output sensor of the simulated CSTR process using the proposed robust FDD scheme

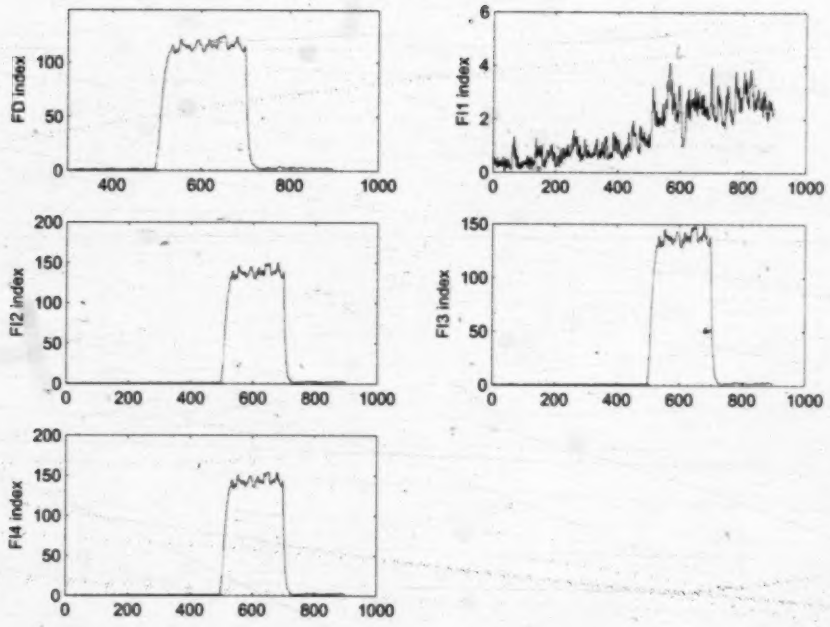

Figure 7.4: Detection and isolation results of a bias fault in the 1^{st} output sensor of the simulated CSTR process using the Chow–Willsky approach

Table 7.2: Singular values of matrix $\mathbf{G}_s^\circ$ and their cumulative percentages

#	Singular Values	Cumulative percentages
1	2.6811	42.86%
2	1.4352	65.80%
3	1.3936	88.08%
4	0.7460	100%

Actuator FDD for the CSTR

Using this numerical example, we also carry out an actuator FDD. First, a PRV insensitive to the first left singular vector of $\mathbf{G}_s^\circ$ i.e. $n_0 = 1$ is designed. The singular values of $\mathbf{G}_s^\circ$ are displayed in Table 7.2, where the first singular value accounts for 42.86% of the total variance of the process uncertainty vector $\mathbf{e}_{s-1}^*(k-1)$. Therefore, 42.86% of the process uncertainties has been removed from the PRV. The dimension of the PRV in this case is 9. We choose L_m to be equal to $3\max(\|\mathbf{B}_\circ \bar{\mathbf{u}}(k)\|) = 0.363$. Consequently, the fault detection index is a non-central chi-square random variable with 3 degrees of freedom and noncentrality parameter 0.6756. With $\alpha = 0.01$, the confidence limit for the fault detection index is 13.6104. Further, from the PRV, two SRVs are generated for actuator fault isolation, which are made insensitive to faults in the first and second actuators, respectively. Similarly, we can define two fault isolation indices: $\{\eta_s^1(k),\ \eta_s^2(k)\}$ which are associated with the SRVs.

A bias fault $f^i(k) = 1.3$ is introduced to one of the two actuators. The FDD results are displayed in Figure 7.5, where the FDD indices have also been scaled to have unit confidence limits. As clearly shown in the figure, because the detection index is beyond its confidence limit after the occurrence of a fault, fault detection is successfully achieved,. Moreover, because the first isolation index is within its confidence limit and the second one is not, according to the pre-determined isolation logic we can conclude that the first actuator has a fault. Note that in the figure, FD stands for the scaled fault detection index, and FI_j, $j = 1, 2$, stand for the scaled fault isolation indices.

Because we know the state and the disturbance vectors, we can check the validity of the detectability condition for actuator faults. In this simulation example, $\max(\|\mathbf{M}\mathbf{e}_{s-1}^*(k-1)\|) = 0.0598$. Thus, the maximum boundary-to-disturbance ratio

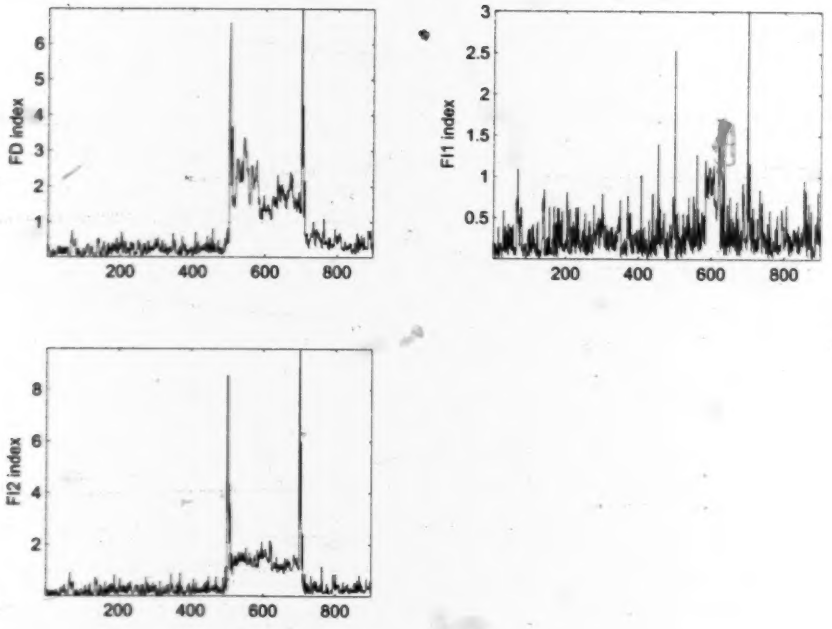

Figure 7.5: Detection and isolation results of a bias fault in the 1^{st} actuator of the simulated CSTR process using the proposed robust FDD approach

is 13.7441. On the other side, $\|\mathbf{l}_{ns} + FDR(k)\| = 13.8062$. Therefore, we can verify that the detectability condition of actuator faults is satisfied.

7.5.2 Experimental Case Study

In this experimental case study, a real continuous stirred tank heater (CSTH) is used for sensor FDD. The CSTH system is located in the Computer Process Control Laboratory, Department of Chemical and Materials Engineering, University of Alberta, Canada.

The CSTH system has (1) two inputs: cold water (CW) and steam (S); (2) four measured outputs: cold water flow rate (F_C), water level (L), and two outlet water temperatures (T_1 and T_2); and (3) one major disturbance: hot water (HW). The cold water and hot water are mixed thoroughly in the tank while being heated by high pressure steam passing through a coil. The water level is controlled by a cold water valve using a PID controller. Two outlet water temperature sensors are located at different locations on a long outlet pipeline. Even though both sensors measure the same physical variable, their readings are not identical due to the different time delays and heat losses. The inputs and outputs vectors are $\mathbf{u} = [CW \ S]'$ and $\mathbf{y} = [L \ F_C \ T_1 \ T_2]'$ respectively, and are sampled every 5 seconds. The overall system is depicted as a block diagram in Figure 7.6.

Using this pilot plant arrangement, a training data set of 400 points is first generated. Subsequently, the following second-order discrete-time state space model is identified from the data by using the *N4SID* function in *Matlab*®, where

$$\mathbf{A}_\circ = \begin{bmatrix} 0 & 1 \\ -0.7494 & 1.6504 \end{bmatrix}, \quad \mathbf{B}_\circ = \begin{bmatrix} -0.1652 & -0.0011 \\ -0.2727 & 0.0029 \end{bmatrix},$$

$$\mathbf{C}_\circ = \begin{bmatrix} -1.1527 & 1.5935 \\ 0.6940 & -3.8102 \\ 1.0125 & -0.0086 \\ 1 & 0 \end{bmatrix}$$

Selecting $s = n = 2$ and using the same procedure as in the numerical example, we generate a PRV for fault detection and four SRVs for fault isolation, where the isolation logic in Table 7.1 is still used.

A bias type of fault $f^i(k) = 1.0$ is introduced to a sensor between the 100^{th} to 200^{th} sample instants. Figure 7.7 illustrates the relevant FDD result, where the fault

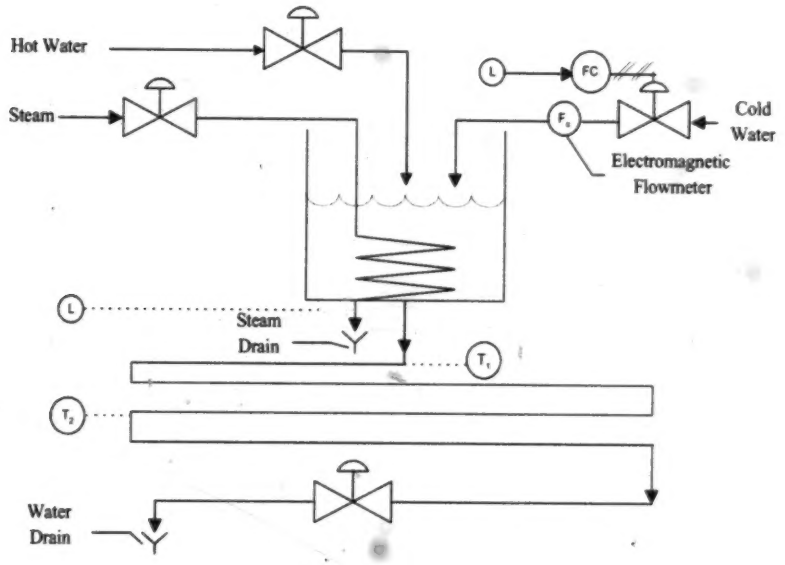

Figure 7.6: Process schematic of the experimental CSTH system

detection index and fault isolation indices are also scaled to have unit confidence limit. In the figure, because the fault isolation indices have an incidence code [1 1 0 1], the fault is detected immediately after it occurs. In addition, it can be inferred that the 3^{rd} output, i.e. the first temperature sensor, is faulty.

The Chow–Willsky approach is also applied to this case study. The corresponding FDD result is shown in Figure 7.8, where due to the presence of the process uncertainties, the fault is incorrectly detected and furthermore erroneously isolated.

7.6 Conclusions

A robust scheme for the detection and isolation of sensor/actuator faults in dynamic processes has been proposed in this chapter. In the presence of sensor faults, this approach can perfectly decouple the effect of any process uncertainties, including MPM and unmeasured disturbances from the PRV, if the number of process uncertainties is less than the number of outputs. In the presence of actuator faults, the principal components of the process uncertainties can be removed from the PRV. Therefore, a robust actuator FDD can also be achieved.

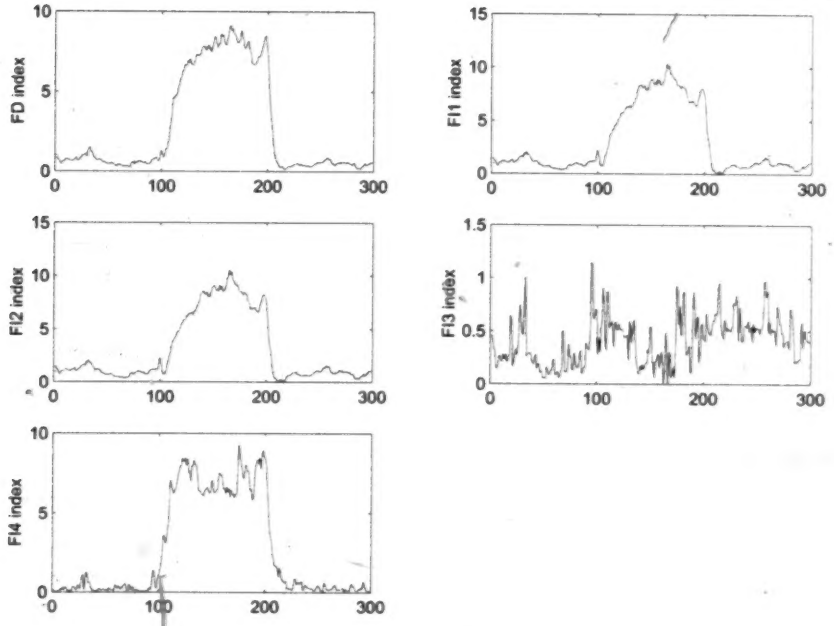

Figure 7.7: Detection and isolation of a bias fault in the 3^{rd} output sensor of the experimental CSTH system using the proposed robust FDD approach

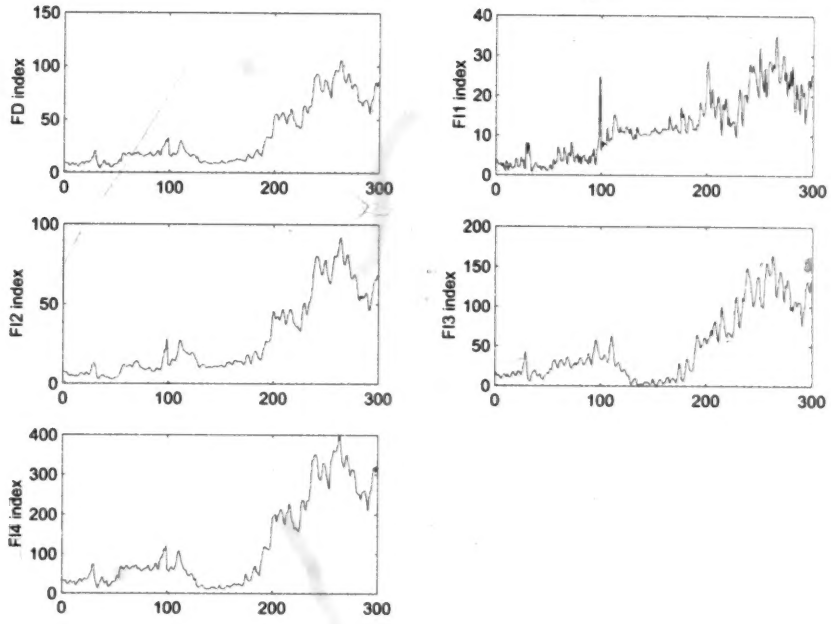

Figure 7.8: Detection and isolation results of a bias fault in the 3^{rd} output sensor of the experimental CSTH system using the Chow–Willsky approach

This newly proposed method is applied to a simulated CSTR system and an experimental equipment. In both cases, satisfactory FDD results are obtained. In addition, in both cases comparisons between the newly proposed FDD method with the Chow–Willsky approach were made. It is demonstrated that the newly proposed FDD approach has acceptable robustness with respect to process uncertainties and a high sensitivity to faults.

8

Detection and Diagnosis of Multiplicative Faults Using Data Reconciliation [5]

8.1 Introduction

Data reconciliation (DR), which has been well studied in recent years ([22] [5] [114]), yields estimators of process variables that satisfy physical constraints of the process. However, to obtain accurate estimates, the influence of gross errors must be eliminated. Hypothesis testing has been extensively used for detection, isolation and identification of gross errors. Gross error detection (GED) is closely related to DR.

[5] A condensed version of this chapter was presented at the *7th International Symposium on DYnamics and COntrol of Process Systems (DYCOPS7)*, July 2004, Boston, USA. It appears in the Proceedings of the conference under the title "Detection and diagnosis of data reconciliation problems in an industrial chemical inventory system" by Han, Shah, Narasimhan and Zaknoun.

Literature surveys of GED can be found in Mah (1990, [79]), Madron (1992, [27]), Sánchez and Romagnoli (2000, [116]) and Narasimhan and Jordache (2000, [114]).

8.2 Multiplicative Fault Detection and Isolation

Suppose the fault-free process measurement vector $\mathbf{z}(k) \in \Re^{m+l}$ can be described by:

$$\mathbf{z}(k) = \tilde{\mathbf{z}}(k) + \sigma(k) \tag{8.1}$$

where $\tilde{\mathbf{z}}(k) \in \Re^{m+l}$ is the vector of true values at time k and $\sigma(k) \in \Re^{m+l}$ is a zero mean normal distributed random vector representing the measurement noise, i.e. $\sigma(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_\sigma)$ where $\mathbf{R}_\sigma \in \Re^{(m+l)\times(m+l)}$ is the covariance matrix of noise $\sigma(k)$. In most cases, the noises are mutually independent, i.e. $\mathbf{R}_\sigma$ is a diagonal matrix.

Assume the process model can be represented by

$$\mathbf{M}^* \tilde{\mathbf{z}}(k) = \mathbf{0} \tag{8.2}$$

where $\mathbf{M}^* \in \Re^{w\times(m+l)}$ is a known matrix and w is the number of constraints.

If sensor(s) have calibration problems, their measurements may differ from $\tilde{\mathbf{z}}$. Different types of calibration errors are illustrated in Figure 8.1. In this chapter, we consider the most general case – bias plus slope error – described mathematically as follows:

$$z_i(k) = \alpha_i \tilde{z}_i(k) + \beta_i + \sigma_i(k) \tag{8.3}$$

where α_i and β_i are two calibration parameters and $\tilde{z}_i$ denotes the i^{th} element of vector $\tilde{\mathbf{z}}$. In the fault-free case, $\alpha_i = 1$ and $\beta_i = 0$.

To directly detect the sensor calibration error, one can apply the measured variables $\mathbf{z}(k)$ to the process constraints and thereby determine whether the residual $\mathbf{r}(k) = \mathbf{M}^* \mathbf{z}(k)$ is approximately zero. Obviously, if sensors do not have calibration errors, the residual is only related to the measurement noise, i.e. $\mathbf{r}(k) = \mathbf{M}^* \sigma(k)$. Therefore, the residual must also be zero-mean random noise. However, if some of the sensors have calibration error, the residual is no longer zero-mean. Thus, one can detect the possible sensor calibration error by examining the mean of $\mathbf{r}(k)$.

Further, one must determine which sensor(s) are the problematic one(s). In this step, one must categorize the process variables into two sets: suspected $\tilde{\mathbf{z}}^s(k) \in \Re^{m_1}$ and unsuspected $\tilde{\mathbf{z}}^u(k) \in \Re^{m_2}$. Thus, the process model $\mathbf{M}^*$ can be rearranged as

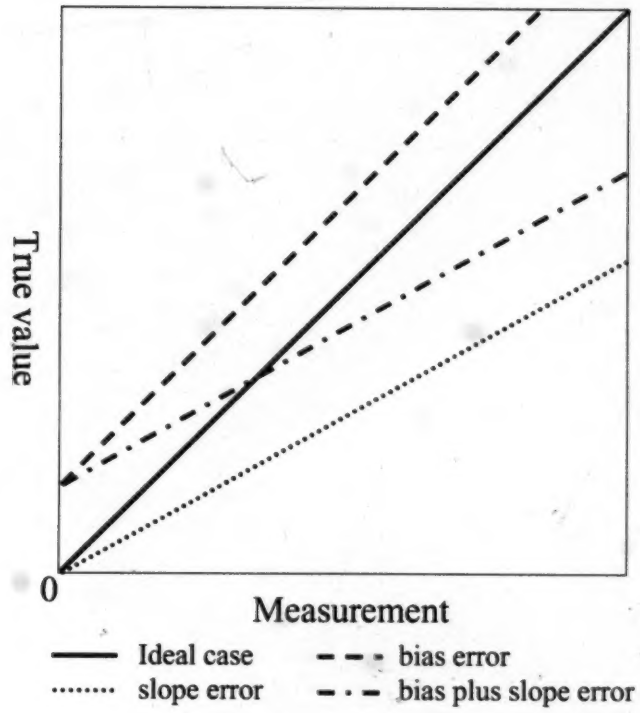

Figure 8.1: Illustration of different types of calibration errors

$\mathbf{M}^* = [\mathbf{M}_1^* \ \mathbf{M}_2^*]$, where $\mathbf{M}_1^*$ and $\mathbf{M}_2^*$ correspond to $\tilde{\mathbf{z}}^s$ and $\tilde{\mathbf{z}}^u$ respectively. Therefore, Eqn. 8.2 can be rewritten as:

$$\mathbf{M}_1^* \tilde{\mathbf{z}}^s(k) + \mathbf{M}_2^* \tilde{\mathbf{z}}^u(k) = \mathbf{0} \tag{8.4}$$

with the assumptions of

$$\begin{aligned} \mathbf{z}^s(k) &= \Lambda \tilde{\mathbf{z}}^s(k) + \Delta + \sigma^s(k) \\ \mathbf{z}^u(k) &= \tilde{\mathbf{z}}^u(k) + \sigma^u(k) \end{aligned}$$

where $\Lambda \in \Re^{m_1 \times m_1}$ is an unknown diagonal matrix and $\Delta \in \Re^{m_1}$ is an unknown vector.

When one applies the above assumptions to the process constraints as depicted by Eqn. 8.4, the following can be obtained:

$$\begin{aligned} \mathbf{r}(k) &= \mathbf{M}_2^* \mathbf{z}^u(k) + \mathbf{M}_1^* \Lambda^{-1} \mathbf{z}^s(k) - \mathbf{M}_1^* \Lambda^{-1} \Delta \\ &= \mathbf{M}_1^* \Lambda^{-1} \sigma^s(k) + \mathbf{M}_2^* \sigma^u(k) \end{aligned} \tag{8.5}$$

Now, the entire problem can be formulated as a *least-squares optimization* problem:

$$J = \min_{\Lambda, \Delta} \mathbf{r}'(k) \mathbf{r}(k) \tag{8.6}$$

Note that only $2m_1$ parameters must be determined by the optimization procedure, as Λ is a diagonal matrix. By observing the estimated value of the diagonal elements of Λ and Δ, one can determine which sensor(s) are mostly likely to have calibration errors by comparing the estimates with the fault-free case.

8.3 Chemical Tank Inventory Process

The process considered in this chapter is a caustic tank inventory system at Millar Western Forest Products Ltd., Whitecourt, Canada. This process serves as a buffer of supplying caustic usage for the entire plant. A simplified schematic of the process is illustrated in Figure 8.2.

In Figure 8.2, the caustic is delivered by trucks at various frequencies depending on the plant's chemical usage. Thus, the inlet flow rate is not continuous. Moreover, it is not measured. There are 6 independent outlet flow control loops. Each pipeline supplies caustic to different parts of the plant, and the flow rate of each outlet pipe is controlled independently.

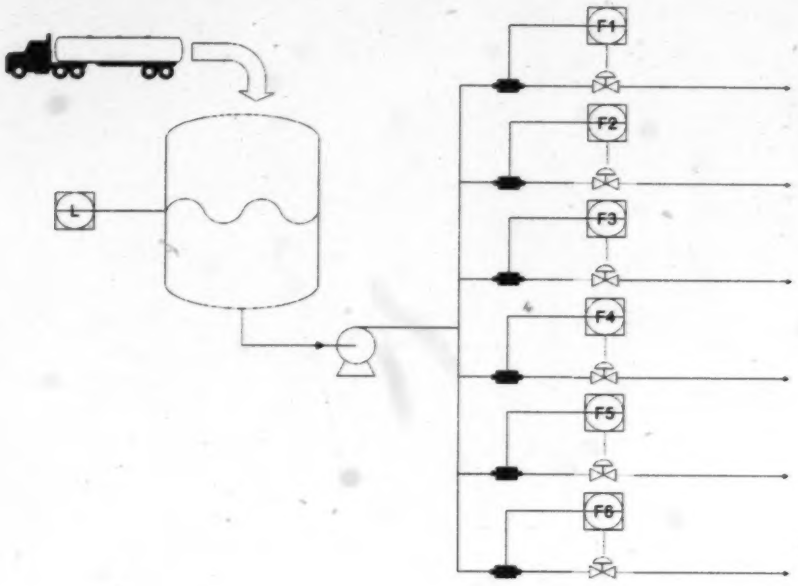

Figure 8.2: Chemical tank inventory process schematics

The level of the tank is not controlled, but is measured. The tank is a vertical cylinder one with a diameter of 6.8 meters and a height of 6.0 meters. Thus the nominal volume is 216 cubic meters. The level sensor is located 0.45 meters above the tank bottom. Therefore, the measurable volume of the tank is about 200 cubic meters. All relevant measured variables are listed in Table 8.1.

From Table 8.1, it is evident that the six flow rate sensors are not using the same unit of measurement. Two of them (F1 and F4) are volume flow sensors (L/min), while the other four are mass flow sensors (kg/min). In this chapter, we assume the delivered caustic has a constant density of 1.52 kg/L. Thus, one can convert each unit into the other.

All six valves are equal percentage valves, which are essentially nonlinear valves. The characteristics of equal percentage valve can be found in the literature ([109] [83]). The system is sampled once every 1 minute, i.e. $T_s = 1\ min.$

Table 8.1: Measured variables of chemical tank inventory system

Name	Description	Units
L	Tank level	%
F1	Flow rate in outflow #1	L/min
F2	Flow rate in outflow #2	kg/min
F3	Flow rate in outflow #3	kg/min
F4	Flow rate in outflow #4	L/min
F5	Flow rate in outflow #5	kg/min
F6	Flow rate in outflow #6	kg/min
P1	Valve position of outflow #1	%
P2	Valve position of outflow #2	%
P3	Valve position of outflow #3	%
P4	Valve position of outflow #4	%
P5	Valve position of outflow #5	%
P6	Valve position of outflow #6	%

8.4 Monitoring System Design and Diagnosis Results

8.4.1 Process Model Description

Mass Balance

The relationships between the volume of the caustic in the tank and the inlet and outlet flow rates can be represented by the following equation:

$$\frac{dV(t)}{dt} = F_{in}(t) - F_{out}(t) \tag{8.7}$$

where $V(t)$ is the chemical volume in the tank, and $F_{in}(t)$ and $F_{out}(t)$ are the inlet and outlet flow rates respectively.

In discrete time domain, the above equation can be written as:

$$r_{MB}(k) = V(k) - V(k-1) - (F_{in}(k) - F_{out}(k))\,T_s = 0 \tag{8.8}$$

where $r_{MB}(k)$ is defined as "mass balance residual". This discrete-time representation implies that the inlet and outlet flow rates do not change between samples. In this process, because the sampling interval is relatively large, this assumption usually cannot be fully satisfied. Therefore, when this discrete-time model is used, a small deviation from zero can be expected due to this systematic model-plant mismatch (MPM).

$F_{out}(k)$ can be easily calculated by summing all the outlet flow rates. $V(k)$ can be obtained from the tank level L and the *a priori* knowledge of the total volume of the storage tank, i.e. $V(k) = 200 \times L(k)\ m^3$. However, $F_{in}(k)$ is not measured in the process. Therefore, according to the essential elements of process monitoring specified in Chapter 2, the process measurements are not redundant.

This does not mean we cannot design a monitoring scheme for the system under consideration. The operation can be categorized into two states – "with delivery" and "without delivery" – with different F_{in}s as below:

$$F_{in}(k) = \begin{cases} 0, & \text{without delivery} \\ \text{unknown}, & \text{with delivery} \end{cases} \tag{8.9}$$

Thus, during the period of "without delivery", we know F_in precisely without measurement. Therefore, when there is no delivery, the redundancy condition is satisfied. We can use this redundancy for the purpose of process monitoring. The details will be shown later in this chapter.

Valve Characteristics

All the control valves in this project are equal percentage valves. Theoretically, the flow rate and valve position have the following relationship ([57]):

$$F = F_{min} R^{S/S_{max}} \tag{8.10}$$

where $R = \frac{F_{max}}{F_{min}}$ is referred to as *rangeability*, F_{max} and F_{min} are the maximum and minimum flow rates respectively, F is flow rate through the valve, and S and S_{max} are the stem position and maximum stem position respectively.

By taking the logarithm for Eqn. 8.10, we can obtain:

$$\log F = \log R \cdot \frac{S}{S_{max}} + \log F_{min} \tag{8.11}$$

Doing so makes one to transfer the nonlinear equal percentage valve characteristic into linear framework. Both the flow rate F and the valve position S/S_{max} are measured in this system. Therefore, the measurements are redundant with respect to the valve characteristic. The unknown parameters $\log R$ and $\log F_{min}$ can be estimated by applying least square estimation to collected data. Due to space limitations, the details of this procedure are omitted.

The models of control valves are different from the mass balance model in this project. Unlike mass balance, which holds true precisely at any point of time, valve characteristics change with wear and tear, and thus cannot be assumed true all the time. In addition, the unknown parameters such as $\log R$ and $\log F_{min}$ must be estimated from data, which can introduce estimation errors. Because of these differences, we must treat these two types of models differently in the subsequent analysis.

8.4.2 Offline Analysis for Sensor Calibration Error

After selecting data for "without delivery", Eqn. 8.8 can be further manipulated into the following:

$$r_{MB}(k) = V(k) - V(k-1) + F_{out}(k)T_s \tag{8.12}$$

Thus, we can obtain the mass balance residual at each time instant illustrated as in Figure 8.3. Note that the residual is not stationary over time, and is usually negative. This problem may be due to flow sensor calibration. Our goal is to determine which sensor(s) have calibration problem and then to estimate the magnitude of the problem.

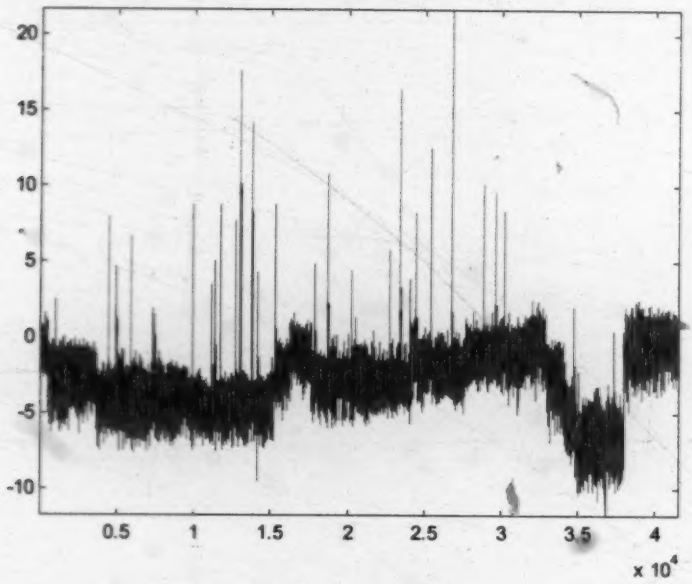

Figure 8.3: Mass balance residual using the points without delivery

In order to confirm that a calibration problem exists, we assume that the level sensor is accurate, but that the flow sensors may have bias plus slope calibration errors, i.e. $F_i^f(k) = \alpha_i F_i^\circ(k) + \beta_i + o_i(k),\ \forall\ i = 1, \cdots, 6$, where $F_i^f(k)$ and $F_i^\circ(k)$ denote the i^{th} measured and true flow rates respectively, α_i and β_i are constant coefficients, and $o_i(k)$ are zero mean normal distributed measurement noise. If the i^{th} flow rate sensor works properly, then ideally $\alpha_i = 1$ and $\beta_i = 0$. Therefore, the mass balance equation can be further written as

$$\begin{aligned} r_{MB}(k) &= V(k) - V(k-1) + \sum_{i=1}^{6} \frac{T_s}{\alpha_i} F_i^f(k) \\ &\quad - \sum_{i=1}^{6} \frac{\beta_i \cdot T_s}{\alpha_i} = \sum_{i=1}^{6} \frac{T_s}{\alpha_i} o_i(k) \end{aligned} \tag{8.13}$$

Because all the β_i are in the same summation operator, one cannot distinguish between them. Thus, we define $\beta = \sum_{i=1}^{6} \frac{\beta_i}{\alpha_i}$. The estimated coefficients can be shown in Table 8.2.

The "Upper bound" and "Lower bound" provide the 95% confidence limits of these estimates in Table 8.2. The estimate of the second flow sensor does not behave

Table 8.2: Estimated calibration coefficients

	α_1	α_2	α_3	α_4	α_5	α_6	β
Estimates	1.0410	1.7354	1.0437	1.0532	1.0084	1.0478	-0.2708
Upper bound	1.2808	2.0551	1.2129	1.3327	1.1965	1.3356	1.2430
Lower bound	0.8011	1.4157	0.8745	0.7737	0.8203	0.7600	-1.7846

normally because the 95% confidence interval of this estimate does not include 1. In contrast, other flow sensors are likely to be well calibrated because their α_i values are close to 1. The total bias-type calibration error β is negligible.

The afore-mentioned results were confirmed by the process control engineer at Miller Western. The flow rate sensor F2 was changed and calibrated in the range of 0 - 15 L/min. However, the conversion range in the DCS system is maintained at the original 0 - 25 L/min. This yields the scaling factor of $25/15 = 1.67$, which is close to our estimate 1.73.

8.4.3 Online Monitoring Design and Implementation

In order to provide an online monitoring scheme for the caustic tank inventory process, the following possible abnormal situations are considered:

- *Sensor faults*: The whole process is equipped with 13 sensors. Any (or all) of these sensors can malfunction during the operation.
- *Process faults*: This type of fault includes any malfunction of process equipment other than instruments, e.g. tank leakage.

To accurately detect and isolate the afore-mentioned faults, analytical redundancy must be used. In this project, there are a total of 7 equations representing the relationships among measured variables: one mass balance and 6 valve characteristic equations. For detection purpose, if one of these equations is not valid, one can infer that something in the system is malfunctioning, i.e. a fault is detected. For isolation purpose, because different types of faults affect different equations, one can determine which one is the root cause by observing which equations are affected. Table 8.3 illustrates how different faults affect different equations.

Concerning Table 8.3, the following remarks can be made:

- The rows denote the possible faults.

Table 8.3: Fault isolation logic

	MB	V_1	V_2	V_3	V_4	V_5	V_6
L	×						
F1	×	×					
F2	×		×				
F3	×			×			
F4	×				×		
F5	×					×	
F6	×						×
P1		×					
P2			×				
P3				×			
P4					×		
P5						×	
P6							×
Process	×						

- The columns denote different equations. MB stands for mass balance and V_i represents the valve characteristic of valve i.
- If there is a $\times$ in the intersection, the corresponding fault affects the corresponding equation. If a blank exists, the fault has no effect on the equation.
- Different faults affect different combinations of equations. Thus, if one or more equations are not valid, one can consult Table 8.3 to determine the root cause of the fault.
- One cannot distinguish between level sensor fault and process faults because they affect the process model identically due to the limited degrees of freedom.

On May 29, 2003, one alarm was generated by the afore-mentioned monitoring scheme. Figure 8.4 depicts the 7 residuals that occurred around that period of time. From Figure 8.4, it can be determined that the valve characteristic equation of flow loop #5 is invalid as well as the mass balance residual. In contrast, the other valve characteristic equations are valid. Therefore, according to the fault isolation logic presented in Table 8.3, it appears that flow sensor #5 (F5) was affected by a measurement problem during that period of time. This online analysis result was confirmed by the process control engineer. Therefore, in this way, a fault in the tank inventory process cannot only be detected but also isolated online.

8.5 Conclusion

In this chapter, the detection and diagnosis of sensor calibration error is proposed and then applied to the caustic tank inventory monitoring project at Millar Western Forest Products Ltd. The proposed method uses a first order model to represent the calibration error and then applies an optimization procedure to solve the problem. The effectiveness of this method was proven through an application. In addition, an online monitoring scheme is designed and implemented by utilizing the process model structure presented in Table 8.3. The effectiveness of this online monitoring scheme is also proved.

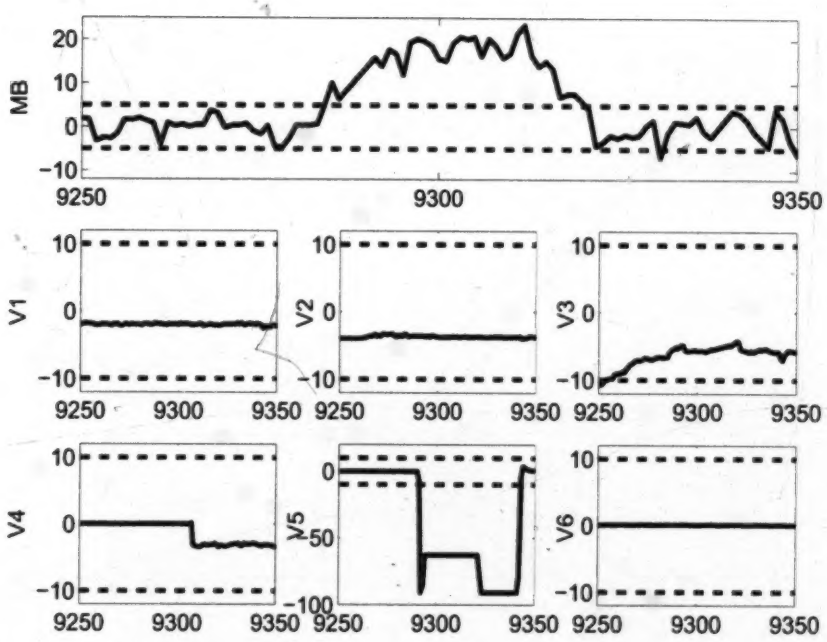

Figure 8.4: Residuals when an alarm is generated

9

Fault Detection and Diagnosis with Multirate Data[6]

9.1 Introduction

In most of the literature on identification, control and fault detection, the standard assumption is that in a given system all input and output data are sampled at a *single* and *uniform* (*regular*) rate, i.e. all variables are sampled *synchronously* and the sampling intervals for each variable are *equally spaced*. In real-world industrial situations, this is often not the case. Frequently, input and output data are sampled *asynchronously*, and sampling is sometimes *non-uniform* (*irregular*). This is so

[6]A version of this chapter has been accepted for publication in *Automatica* under the title "Subspace Identification for FDI in Systems with Non-uniformly Sampled Multirate Data" by Li, Han and Shah.

primarily because there are delays in sensors and laboratory analysis. For example, consider a polymer reactor in the chemical industry, where the composition, density, and molecular weight distribution measurements are typically obtained after several minutes of analysis, while the manipulated variables are often adjusted at relatively fast rates ([94], [43]). Common example of sampling at multiple and non-uniform rates is illustrated in Figure 9.1.

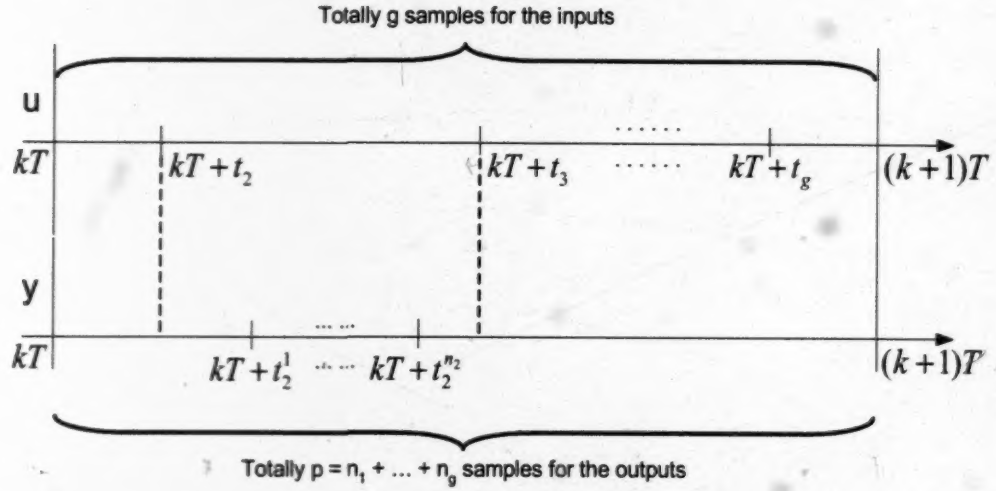

Figure 9.1: Schematic representation of a process where variables are non-uniformly sampled at different rates

The primary reason for considering a non-uniform and multirate sampling framework for identification with or without fault detection is that all other uniformly sampled systems are special cases of this general framework. In addition to its practicality, non-uniform sampling preserves causality automatically, and is both controllable and observable under a non-strict condition in discretization ([119]).

While an enormous amount of research effort has been dedicated at fault detection and diagnosis (FDD) in systems with single rate data, only limited work has been reported for FDD in systems with multirate data ([29],[30], [28],[135], [69]).

Existing studies assume knowledge of a continuous-time (CT) model of the system. By *lifting*, a discrete-time (DT) model of the system is obtained from the CT model. Then, based on the *lifted* DT model, residual models are designed and manipulated for FDD. Lifting transforms a time-varying multirate system into a time-invariant single rate system. Due to the work by Khargonekar *et al.* ([62]), lifting has become

a standard tool for dealing with multirate systems.

A CT model of a physical system is typically obtained from a first principle model. Because establishing a first principle model for a system depends on fully understanding the mechanism of the system, such a model is not always easily achievable or practical for a complex industrial system. If a mechanistic model is unavailable, then one cannot derive the *lifted model* of the system. As a consequence, a residual model cannot be designed for FDD.

This chapter extends the Chow-Willsky scheme ([19]) - originally proposed for single rate data - to FDD in systems with non-uniform sampled multirate (NUSM) data. In particular, the chapter investigates direct identification of residual models from training data to generate a primary residual vector (PRV) for fault detection and a set of structured residual vectors (SRVs) for fault isolation.

Consider a physical system that is represented by a state space model in the CT domain. The multirate non-uniformly sampling technique proposed by Sheng *et al.* ([119]) is used to collect data from such a system and then to obtain a *lifted system* represented by a state space model, e.g. $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$. Conventionally, one first identifies $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ by means of SIM, and then calculates the residual models for FDD with the identified matrices.

SIM were initially proposed for the identification of state space models in single rate DT systems ([86], [96],[97], [18]), but have now been extended to uniform multirate systems ([68]). If the residual models are derived from the identified $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, all errors in the identification step become lumped.

As will be shown later, to generate the residual models for FDD, one does not need the individual values of $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$. Instead, one needs only an extended observability matrix, $\mathbf{\Gamma}_s$, and a lower triangular block Toeplitz matrix, $\mathbf{H}_s$, both of which are functions of $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$. The objective of this chapter is to develop a subspace algorithm for direct identification of $\mathbf{\Gamma}_s$ and $\mathbf{H}_s$ from NUSM data without identifying $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$. It is considered to be an important FDD-relevant identification step.

This chapter is organized as follows. The problem is formulated in Section 2. In Section 3, a novel technique is developed to derive a single rate linear time-invariant (LTI) model, i.e. a *lifted* model, for a system with NUSM data by *integration*. Identification of residual models for fault detection is investigated in Section 4, wherein the optimal design of a PRV is also outlined. In addition, a numerically-robust identification algorithm is developed. An FDD case study is given in Section 5, where we use data collected from an experimental pilot-scale plant to demonstrate

the effectiveness of the proposed scheme. Design of residual models for SRVs is included. Concluding remarks are presented in Section 6. Detailed derivations of several equations are provided in Appendices 9.A-9.D.

9.2 Problem Formulation

Assume that a dynamic system in the *fault-free* case is represented by the following CT state space equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t),\ \tilde{\mathbf{y}}(t) = \mathbf{C}_c\mathbf{x}(t) + \mathbf{D}_c\tilde{\mathbf{u}}(t) \tag{9.1}$$

where (1) $\tilde{\mathbf{u}}(t) \in \Re^l$ and $\tilde{\mathbf{y}}(t) \in \Re^m$ are *noise-free* inputs and outputs respectively; (2) $\mathbf{x}(t) \in \Re^n$ is the state; (3) $\boldsymbol{\phi}(t) \in \Re^n$ is a Gaussian distributed white noise vector with covariance $\mathbf{R}_\phi$; and (4) $\mathbf{A}_c$, $\mathbf{B}_c$, $\mathbf{C}_c$ and $\mathbf{D}_c$ are unknown system matrices with appropriate dimensions. It is further assumed that (1) the pair $(\mathbf{A}_c, \mathbf{C}_c)$ is observable, (2) the pair $(\mathbf{A}_c, \mathbf{B}_c\mathbf{R}_\phi^{1/2})$ is controllable, and (3) the stochastic portion of $\mathbf{A}_c$ is asymptotically stable.

The non-uniform multirate sampling approach ([119]) is applied to obtain measurements from Eqn. 9.1. More specifically, for a given *frame period*, T, over the k^{th} frame period $[kT, kT+T)$, we sample the inputs and outputs non-uniformly as follows.

- An input variable is sampled g times at time instants: $\{kT+t_1, kT+t_2, kT+t_3, \cdots, kT+t_g\}$. Without loss of generality, we arrange $0 = t_1 < t_2 < \cdots < t_g < T$.
- An output variable is sampled p times. Moreover, within the time interval $[kT+t_i, kT+t_{i+1})$, for $i = [1, ..., g]$, n_i (≥ 0) samples of the output variable are taken at time instants: $\{kT+t_i^1, kT+t_i^2, \cdots, kT+t_i^{n_i}\}$. Similarly, we arrange $t_i \leq t_i^1 < t_i^2 \cdots < t_i^{n_i} < t_{i+1}$, where $t_{g+1} = T$. Note that $p = n_1 + n_2 + \cdots + n_g$ can be larger/smaller than, or equal to, g. When $n_i = 0$, no sample is taken within the interval $[kT+t_i, kT+t_{i+1})$.

The sampling is repeated over the next frame period.

In the most general case, among the $m+l$ inputs and outputs, each variable is sampled differently from the others. However, for simplicity of mathematical presentation and manipulation, without loss of generality we assume that (1) the l inputs, $\tilde{\mathbf{u}}(t)$, and the disturbances are sampled at one rate; and (2) the m outputs,

$\tilde{\mathbf{y}}(t)$, are sampled at another rate. Such non-uniform and multirate sampling is illustrated in Figure 9.2.

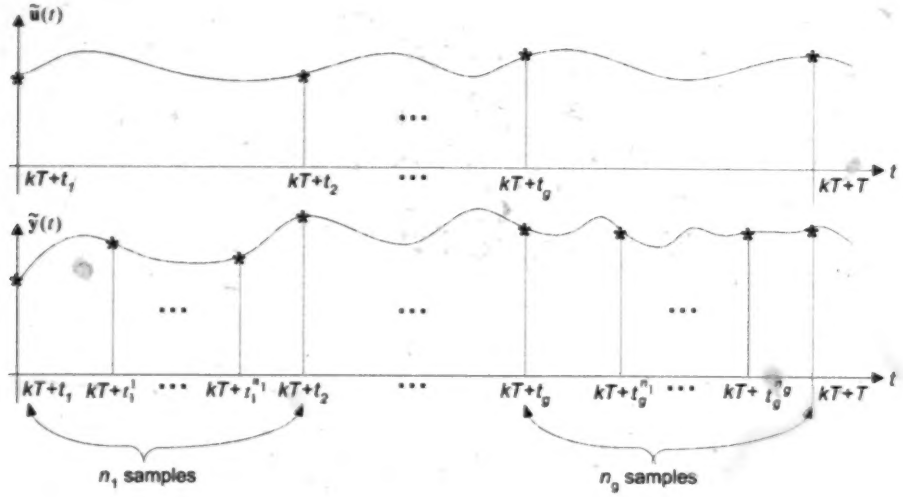

Figure 9.2: Non-uniform and multirate sampling of inputs and outputs

Consider the case of errors-in-variables, and denote the observed *fault-free* inputs at the time instant $kT + t_i$, for $i = [1, ..., g]$, by $\mathbf{u}(kT + t_i) = \tilde{\mathbf{u}}(kT + t_i) + \mathbf{v}(kT + t_i)$. Similarly, at the time instant $kT + t_i^j$, for $j = [1, ..., n_i]$, the fault-free outputs are denoted by $\mathbf{y}(kT+t_i^j) = \tilde{\mathbf{y}}(kT+t_i^j)+\mathbf{o}(kT+t_i^j)$, where $\mathbf{v}$ and $\mathbf{o}$ are measurement errors and are assumed to be Gaussian distributed white noise vectors with corresponding covariance matrices $\mathbf{R}_v$ and $\mathbf{R}_o$. Further, it is assumed that $\mathbf{v}$ and $\mathbf{o}$ are independent of the initial state $\mathbf{x}(0)$, and are mutually independent.

If sensors are faulty, their measurements include *fault-free* and *fault-related* values. Therefore, the measured outputs with sensor faults, for $j = [1, ..., n_i]$, can be represented by

$$\mathbf{y}(kT + t_i^j) = \tilde{\mathbf{y}}(kT + t_i^j) + \mathbf{o}(kT + t_i^j) + \mathbf{f}_y(kT + t_i^j) \tag{9.2}$$

where $\mathbf{f}_y(kT+t_i^j) \in \Re^m$ is the fault magnitude vector with zero and non-zero elements. To represent a single sensor fault in the i^{th} output sensor, for $i = [1, ..., m]$, the i^{th} element of $\mathbf{f}_y(kT + t_i^j)$ is non-zero, but all other elements are zero. Moreover, to represent simultaneous multiple sensor faults in the outputs, e.g. faults in the second and the fourth output sensors, the second and the fourth elements of $\mathbf{f}_y(kT + t_i^j)$ are non-zero, but all other elements are zero. The measured inputs can be represented

by

$$\mathbf{u}(kT+t_i) = \tilde{\mathbf{u}}(kT+t_i) + \mathbf{v}(kT+t_i) + \mathbf{f}_u(kT+t_i) \tag{9.3}$$

where $\mathbf{f}_u(kT+t_i)$ is the fault vector in the input sensors and structurally similar to $\mathbf{f}_y(kT+t_i)$.

In the case of the errors-in-variables, this chapter considers solutions to the following three problems:

- Given measured training data: $\{\mathbf{u}(kT+t_i)\}$ and $\{\mathbf{y}(kT+t_i^j)\}$, for $i = [1,...,g]$, $j = [1,...,n_i]$, and $k = [1,2,\cdots)$, which are collected while the considered system is fault-free, residual models can be identified.
- With identified residual models and test data: $\{\mathbf{u}(kT+t_i)\}$ and $\{\mathbf{y}(kT+t_i^j)\}$, for the same i, j, and an identical or different k, fault detection can be performed, i.e. to indicate when $\mathbf{f}_u(kT+t_i)$ and/or $\mathbf{f}_y(kT+t_i^j)$ are non-zero.
- After fault detection, the faulty sensors can be identified. This is referred to as *fault isolation*.

9.3 The Lifted Model for the System with NUSM Data

When inputs and outputs in the CT system described by Eqn. 9.1 are non-uniformly sampled at multirates, the result is a time-varying system. However, if we group all g input measurements and all p output measurements together, we will have a single rate LTI model with an increased dimension for the system. Herein we refer to $\tilde{\mathbf{u}}(k) = \tilde{\mathbf{u}}(t)|_{t=kT}$ as a *vectored-input measurement*, and similarly we refer to $\tilde{\mathbf{y}}(k) = \tilde{\mathbf{y}}(t)|_{t=kT}$ as a *vectored-output measurement*. This terminology will be used throughout the chapter.

Sheng et al. ([119]) have shown the derivation of a lifted model for a system with NUSM data by using the conventional lifting technique. This chapter derives the lifted model in a novel way by means of *integration*. The fundamental concept of lifting a system is to approximate the integrals of functions of system variables within each frame period under certain assumptions. Such a derivation of a lifted model from an integration perspective is more natural, straightforward, and understandable than the conventional one.

Post-multiplying the first line of Eqn. 9.1 by $\mathbf{e}^{-\mathbf{A}_c t}$ (assumed to be non-singular for any finite t) leads to $\mathbf{e}^{-\mathbf{A}_c t}\dot{\mathbf{x}}(t) = \mathbf{e}^{-\mathbf{A}_c t}\mathbf{A}_c\mathbf{x}(t) + \mathbf{e}^{-\mathbf{A}_c t}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t)\right]$, or equivalently,

$$\frac{d\left[\mathbf{e}^{-\mathbf{A}_c t}\mathbf{x}(t)\right]}{dt} = \mathbf{e}^{-\mathbf{A}_c t}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t)\right] \tag{9.4}$$

In Eqn. 9.4, we have employed $\mathbf{e}^{-\mathbf{A}_c t}\dot{\mathbf{x}}(t) - \mathbf{e}^{-\mathbf{A}_c t}\mathbf{A}\mathbf{x}(t) = d\left(\mathbf{e}^{-\mathbf{A}_c t}\mathbf{x}(t)\right)/dt$, where $d(\,)/dt$ stands for the derivative of the argument with respect to t.

Integrating Eqn. 9.4 from $t = kT$ to $t = kT + T$ gives

$$\mathbf{e}^{-\mathbf{A}_c(kT+T)}\mathbf{x}(kT+T) - \mathbf{e}^{-\mathbf{A}_c kT}\mathbf{x}(kT) = \int_{kT}^{kT+T} \mathbf{e}^{-\mathbf{A}_c t}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t)\right] dt,$$

Multiplying both sides by $\mathbf{e}^{\mathbf{A}_c(kT+T)}$ leads to

$$\mathbf{x}(kT+T) \;=\; e^{\mathbf{A}_c T}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{A}_c(kT+T-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t)\right] dt.$$

Introducing new conventions: $\mathbf{x}(k) \equiv \mathbf{x}(kT)$, $\mathbf{x}(k+1) \equiv \mathbf{x}(kT+T)$, $\mathbf{A} \equiv \mathbf{e}^{\mathbf{A}_c T}$, we can rewrite the preceding equation as

$$\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \int_{kT}^{kT+T} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t)\right] dt \tag{9.5}$$

By sampling $\tilde{\mathbf{u}}(t)$ for $t \in [kT, kT+T)$, we obtain g samples of inputs. Similarly, we obtain g samples of disturbances by "sampling" $\boldsymbol{\phi}(t)$. Accordingly, we form two lifted vectors:

$$\underline{\tilde{\mathbf{u}}}(k) = \begin{bmatrix} \tilde{\mathbf{u}}(kT+t_1) \\ \tilde{\mathbf{u}}(kT+t_2) \\ \vdots \\ \tilde{\mathbf{u}}(kT+t_g) \end{bmatrix} \in \Re^{lg}, \quad \underline{\boldsymbol{\phi}}(k) = \begin{bmatrix} \boldsymbol{\phi}(kT+t_1) \\ \boldsymbol{\phi}(kT+t_2) \\ \vdots \\ \boldsymbol{\phi}(kT+t_g) \end{bmatrix} \in \Re^{ng}$$

and arrive at the following lifted state space equation,

$$\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\underline{\tilde{\mathbf{u}}}(k) + \mathbf{E}\,\underline{\boldsymbol{\phi}}(k) \tag{9.6}$$

The detailed derivation of Eqn. 9.6 is given in Appendix 9.A.

From the distribution of $\boldsymbol{\phi}$, it can be inferred that $\underline{\boldsymbol{\phi}}(k) \sim \mathcal{N}(\mathbf{0}, \underline{\mathbf{R}}_\phi)$, where $\underline{\mathbf{R}}_\phi = \mathbf{I}_g \otimes \mathbf{R}_\phi$. One can argue that disturbances are usually not measurable, and as a result $\underline{\boldsymbol{\phi}}(k)$ can not be available. In the ensuing discussion we consider the use of $\underline{\boldsymbol{\phi}}(k)$ only for the purpose of mathematical manipulation. As will be shown later, the disturbances will not affect the identification of the residual models.

The lifted output equation can be similarly derived. Integrating Eqn. 9.4 with $t \in [kT, kT+\tau]$ and $0 < \tau < T$ produces

$$\mathbf{x}(kT+\tau) = \mathbf{e}^{\mathbf{A}_c\tau}\mathbf{x}(k) + \int_{kT}^{kT+\tau} \mathbf{e}^{\mathbf{A}_c(kT+\tau-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t)\right] dt \tag{9.7}$$

where τ is a positive number. On the other hand, it turns out from the second term in Eqn. 9.1 that $\tilde{\mathbf{y}}(kT+\tau) = \mathbf{C}_c\mathbf{x}(kT+\tau) + \mathbf{D}_c\tilde{\mathbf{u}}(kT+\tau)$. The use of Eqn. 9.7 in $\tilde{\mathbf{y}}(kT+\tau)$ leads to

$$\begin{aligned}\tilde{\mathbf{y}}(kT+\tau) = {} & \mathbf{C}_c\mathbf{e}^{\mathbf{A}_c\tau}\mathbf{x}(k) + \mathbf{C}_c\int_{kT}^{kT+\tau} \mathbf{e}^{\mathbf{A}_c(kT+\tau-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t) + \boldsymbol{\phi}(t)\right] dt \\ & + \mathbf{D}_c\tilde{\mathbf{u}}(kT+\tau)\end{aligned} \tag{9.8}$$

After defining the lifted output vector as:

$$\underline{\tilde{\mathbf{y}}}(k) \equiv \left[\ \tilde{\mathbf{y}}'(kT+t_1^1)\ \cdots\ \tilde{\mathbf{y}}'(kT+t_1^{n_1})\ \cdots\ \tilde{\mathbf{y}}'(kT+t_g^1)\ \cdots\ \tilde{\mathbf{y}}'(kT+t_g^{n_g})\ \right]' \tag{9.9}$$

One can show that the lifted output equation is

$$\underline{\tilde{\mathbf{y}}}(k) = \mathbf{C}\,\mathbf{x}(k) + \mathbf{D}\,\underline{\tilde{\mathbf{u}}}(k) + \mathbf{J}\,\underline{\boldsymbol{\phi}}(k) \tag{9.10}$$

The derivation of Eqn. 9.10 is provided in Appendix 9.B. Putting Eqns. 9.6 and 9.10 together, we eventually arrive at the lifted model of Eqn. 9.1:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\underline{\tilde{\mathbf{u}}}(k) + \mathbf{E}\,\underline{\boldsymbol{\phi}}(k) \\ \underline{\tilde{\mathbf{y}}}(k) &= \mathbf{C}\,\mathbf{x}(k) + \mathbf{D}\,\underline{\tilde{\mathbf{u}}}(k) + \mathbf{J}\,\underline{\boldsymbol{\phi}}(k)\end{aligned} \tag{9.11}$$

We assume that subsequently in this chapter, the frame period T is non-pathological relative to matrix $\mathbf{A}$, i.e., the difference between any two eigenvalues of $\mathbf{A}$ is not equal to a multiple of $i2\pi/T$ with $i^2 = -1$. Under this assumption, the lifted system described by Eqn. 9.11 preserves the causality, controllability and observability of the original CT system represented by Eqn. 9.1 ([119]).

9.4 Identification of Residual Models for Fault Detection

In this section, it is first shown that the residual models can be derived from two matrices, $\boldsymbol{\Gamma}_s$ and $\mathbf{H}_s$. Then, identification algorithms for the two matrices are proposed.

9.4.1 Description of the Lifted Model with Faults

We define a stacked vector: $\underline{\tilde{\mathbf{y}}}_s(k) \equiv [\underline{\tilde{\mathbf{y}}}'(k-s)\ \ \underline{\tilde{\mathbf{y}}}'(k-s+1)\ \ \ldots \underline{\tilde{\mathbf{y}}}'(k)]' \in \Re^{pm_s}$, where $m_s \equiv m(s+1)$, and for $j = [0, ..., s]$, $\underline{\tilde{\mathbf{y}}}(k-j) \in \Re^{pm}$ has a definition similar to $\underline{\tilde{\mathbf{y}}}(k)$. Note that subsequently in this chapter, stacked vectors are defined analogously to $\underline{\tilde{\mathbf{y}}}_s(k)$.

Further, we define two more stacked vectors $\underline{\tilde{\mathbf{u}}}_s(k)$ and $\underline{\boldsymbol{\phi}}_s(k)$. Manipulation of Eqn. 9.11 in steps similar to those presented by Li and Shah ([70]) yields the following stacked equation:

$$\underline{\tilde{\mathbf{y}}}_s(k) = \boldsymbol{\Gamma}_s \mathbf{x}(k-s) + \mathbf{H}_s \underline{\tilde{\mathbf{u}}}_s(k) + \mathbf{G}_s \underline{\boldsymbol{\phi}}_s(k) \tag{9.12}$$

where s is the order of the parity space [19] and, for the sake of simplicity, is selected to be equal to n. Furthermore, in Eqn. 9.12, $\boldsymbol{\Gamma}_s = [\mathbf{C}'\ \mathbf{A}'\mathbf{C}'\ \ldots (\mathbf{A}^s)'\mathbf{C}']' \in \Re^{pm_s \times n}$ is the extended observability matrix;

$$\mathbf{H}_s = \left[\begin{array}{c|c} \mathbf{D} & \mathbf{0} \\ \hline \mathbf{C}\,\mathbf{B} & \\ \vdots & \mathbf{H}_{s-1} \\ \mathbf{C}\,\mathbf{A}^{s-1}\mathbf{B} & \end{array}\right] \in \Re^{pm_s \times gl_s} \text{ and } \mathbf{G}_s = \left[\begin{array}{c|c} \mathbf{J} & \mathbf{0} \\ \hline \mathbf{C}\,\mathbf{E} & \\ \vdots & \mathbf{G}_{s-1} \\ \mathbf{C}\,\mathbf{A}^{s-1}\mathbf{E} & \end{array}\right] \in \Re^{pm_s \times gn_s}$$

are two lower triangular block Toeplitz matrices with $n_s = n(s+1)$, $l_s = l(s+1)$, $\mathbf{H}_0 = \mathbf{D}$, and $\mathbf{G}_0 = \mathbf{J}$.

When the sampled inputs and outputs contain measurement noise and sensor faults, it follows from Eqns. 9.2 and 9.3 that the lifted vectors of measured inputs and outputs will be

$$\begin{aligned} \underline{\mathbf{u}}(k) &= \underline{\tilde{\mathbf{u}}}(k) + \underline{\mathbf{v}}(k) + \underline{\mathbf{f}}_u(k) \\ \underline{\mathbf{y}}(k) &= \underline{\tilde{\mathbf{y}}}(k) + \underline{\mathbf{o}}(k) + \underline{\mathbf{f}}_y(k) \end{aligned} \tag{9.13}$$

In Eqn. 9.13, $\{\underline{\mathbf{u}}(k), \underline{\mathbf{v}}(k), \underline{\mathbf{f}}_u(k)\}$ and $\{\underline{\mathbf{y}}(k), \underline{\mathbf{o}}(k), \underline{\mathbf{f}}_y(k)\}$ are structurally identical to $\underline{\tilde{\mathbf{u}}}(k)$ and $\underline{\tilde{\mathbf{y}}}(k)$ respectively. In addition, $\underline{\mathbf{o}}(k) \sim \mathcal{N}(\mathbf{0}, \underline{\mathbf{R}}_o)$, $\underline{\mathbf{v}}(k) \sim \mathcal{N}(\mathbf{0}, \underline{\mathbf{R}}_v)$, with $\underline{\mathbf{R}}_o = \mathbf{I}_p \otimes \mathbf{R}_o \in \Re^{pm}$ and $\underline{\mathbf{R}}_v = \mathbf{I}_g \otimes \mathbf{R}_v \in \Re^{gl}$.

Stacking Eqn. 9.13 leads to the relationship between the stacked vectors:

$$\begin{aligned} \underline{\mathbf{u}}_s(k) &= \underline{\tilde{\mathbf{u}}}_s(k) + \underline{\mathbf{v}}_s(k) + \underline{\mathbf{f}}_{s,u}(k) \in \Re^{gl_s} \\ \underline{\mathbf{y}}_s(k) &= \underline{\tilde{\mathbf{y}}}_s(k) + \underline{\mathbf{o}}_s(k) + \underline{\mathbf{f}}_{s,y}(k) \in \Re^{pm_s} \end{aligned} \tag{9.14}$$

Using Eqn. 9.14, one can rewrite Eqn. 9.12 as

$$\underline{\mathbf{y}}_s(k) = \boldsymbol{\Gamma}_s \mathbf{x}(k-s) + \mathbf{H}_s \left[\underline{\mathbf{u}}_s(k) - \underline{\mathbf{v}}_s(k) - \underline{\mathbf{f}}_{s,u}(k)\right] + \mathbf{G}_s \underline{\boldsymbol{\phi}}_s(k) + \underline{\mathbf{o}}_s(k) + \underline{\mathbf{f}}_{s,y}(k),$$

which can be further manipulated into

$$\begin{aligned}\underline{\mathbf{y}}_s(k) - \mathbf{H}_s\underline{\mathbf{u}}_s(k) &= \boldsymbol{\Gamma}_s\mathbf{x}(k-s) - \mathbf{H}_s\underline{\mathbf{f}}_{s,u}(k) + \underline{\mathbf{f}}_{s,y}(k) + \\ &\quad \mathbf{G}_s\underline{\boldsymbol{\phi}}_s(k) - \mathbf{H}_s\underline{\mathbf{v}}_s(k) + \underline{\mathbf{o}}_s(k) \end{aligned} \tag{9.15}$$

Eqn. 9.15 links the stacked faults, disturbances, measurements, noise, and the dynamics of the lifted model together. Three remarks are in order.

Remark 9.4.1 Assuming that $\mathbf{H}_s$ and $\boldsymbol{\Gamma}_s$ are available, one can extend the Chow-Willsky scheme ([19]) to generate a signal vector as follows

$$\begin{aligned}\underline{\mathbf{e}}_s(k) &\equiv \mathbf{W}_o\left[\underline{\mathbf{y}}_s(k) - \mathbf{H}_s\underline{\mathbf{u}}_s(k)\right] \\ &= \mathbf{W}_o\left[\underline{\mathbf{f}}_{s,y}(k) - \mathbf{H}_s\underline{\mathbf{f}}_{s,u}(k)\right] + \mathbf{W}_o\left[\mathbf{G}_s\underline{\boldsymbol{\phi}}_s(k) - \mathbf{H}_s\underline{\mathbf{v}}_s(k) + \underline{\mathbf{o}}_s(k)\right] \end{aligned} \tag{9.16}$$

where, $\mathbf{W}_o$ is a matrix selected from the *left null space* of matrix $\boldsymbol{\Gamma}_s$, i.e. $\mathbf{W}_o\boldsymbol{\Gamma}_s = \mathbf{0}$. In addition, note that in Eqn. 9.16 the unknown state vector $\mathbf{x}(k-s)$ has been completely removed from the PRV.

Remark 9.4.2 Define $\underline{\mathbf{e}}_s(k)$ as a PRV for fault detection, based on the following facts:

- In the ideal case, $\underline{\mathbf{e}}_s(k) = \mathbf{0}$, because $\underline{\boldsymbol{\phi}}_s(k) = \mathbf{0}$, $\underline{\mathbf{o}}_s(k) = \mathbf{0}$, $\underline{\mathbf{v}}_s(k) = \mathbf{0}$, $\underline{\mathbf{f}}_{s,u}(k) = \mathbf{0}$, and $\underline{\mathbf{f}}_{s,y}(k) = \mathbf{0}$.

- In the fault-free case, $\underline{\mathbf{e}}_s(k) = \mathbf{W}_0\left[\mathbf{G}_s\underline{\boldsymbol{\phi}}_s(k) - \mathbf{H}_s\underline{\mathbf{v}}_s(k) + \underline{\mathbf{o}}_s(k)\right]$, which is denoted by $\underline{\mathbf{e}}_s^*(k)$, is a moving average process of disturbance and noise vectors: $\underline{\boldsymbol{\phi}}(k)$, $\underline{\mathbf{o}}(k)$, and $\underline{\mathbf{v}}(k)$, and follows a zero-mean multivariate Gaussian distribution ([58]), i.e. $\underline{\mathbf{e}}_s^*(k) \sim \mathcal{N}\left(\mathbf{0}, \underline{\mathbf{R}}_{s,e}\right)$ with $\underline{\mathbf{R}}_{s,e} = \mathbf{W}_o\left(\mathbf{G}_s\underline{\mathbf{R}}_{s,\phi}\mathbf{G}_s' + \mathbf{H}_s\underline{\mathbf{R}}_{s,v}\mathbf{H}_s' + \underline{\mathbf{R}}_{s,o}\right)\mathbf{W}_0'$. Note that $\underline{\mathbf{R}}_{s,\phi} = \mathbf{I}_{s+1} \otimes \underline{\mathbf{R}}_\phi$, $\underline{\mathbf{R}}_{s,o} = \mathbf{I}_{s+1} \otimes \underline{\mathbf{R}}_o$, and $\underline{\mathbf{R}}_{s,v} = \mathbf{I}_{s+1} \otimes \underline{\mathbf{R}}_v$ are covariances of $\underline{\boldsymbol{\phi}}_s(k)$, $\underline{\mathbf{o}}_s(k)$ and $\underline{\mathbf{v}}_s(k)$ respectively. As will be seen later, $\underline{\mathbf{R}}_{s,e}$ can be directly estimated from the training data.

- In the presence of any sensor faults,

$$\underline{\mathbf{e}}_s(k) = \underline{\mathbf{e}}_s^*(k) + \underline{\mathbf{e}}_s^f(k) \tag{9.17}$$

  where $\underline{\mathbf{e}}_s^f(k) = \mathbf{W}_0\left[\underline{\mathbf{f}}_{s,y}(k) - \mathbf{H}_s\underline{\mathbf{f}}_{s,u}(k)\right]$ is contributed by a fault(s). Clearly, $\underline{\mathbf{e}}_s(k)$ follows a multivariate Gaussian distribution with mean $\underline{\mathbf{e}}_s^f(k)$ and covariance $\underline{\mathbf{R}}_{s,e}$, i.e. $\underline{\mathbf{e}}_s(k) \sim \mathcal{N}\left(\underline{\mathbf{e}}_s^f(k), \underline{\mathbf{R}}_{s,e}\right)$.

Remark 9.4.3 As shown in Eqn. 9.16, the PRV is

$$\underline{\mathbf{e}}_s(k) = \mathbf{W}_o \left[\mathbf{I}_{pm_s} \mid -\mathbf{H}_s\right] \begin{bmatrix} \underline{\mathbf{y}}_s(k) \\ \underline{\mathbf{u}}_s(k) \end{bmatrix}.$$

Therefore, the residual model for the PRV is $\mathbf{M}_s \equiv \mathbf{W}_o \left[\mathbf{I}_{pm_s} \mid -\mathbf{H}_s\right]$. Furthermore, $\mathbf{M}_s$ can be uniquely determined from $\mathbf{\Gamma}_s$ and $\mathbf{H}_s$ as will be illustrated in the following sections. Hence, to obtain $\mathbf{M}_s$, one only needs $\mathbf{\Gamma}_s$ and $\mathbf{H}_s$.

9.4.2 Identification of Γ_s and $\mathbf{H}_s$

The Sample-wise Stacked Equation

In the fault-free case, because $\underline{\mathbf{f}}_{s,u}(k) = \mathbf{0}$ and $\underline{\mathbf{f}}_{s,y}(k) = \mathbf{0}$, Eqn. 9.15 is reduced to

$$\underline{\mathbf{y}}_s(k) - \mathbf{H}_s \underline{\mathbf{u}}_s(k) = \mathbf{\Gamma}_s \mathbf{x}(k-s) + \mathbf{G}_s \underline{\boldsymbol{\phi}}_s(k) - \mathbf{H}_s \underline{\mathbf{v}}_s(k) + \underline{\mathbf{o}}_s(k) \tag{9.18}$$

Eqn. 9.18 can then be extended to

$$\begin{aligned} \underline{\mathbf{y}}_s(k+1+s) - \mathbf{H}_s \underline{\mathbf{u}}_s(k+1+s) &= \mathbf{\Gamma}_s \mathbf{x}(k+1) + \mathbf{G}_s \underline{\boldsymbol{\phi}}_s(k+1+s) \\ &\quad - \mathbf{H}_s \underline{\mathbf{v}}_s(k+1+s) + \underline{\mathbf{o}}_s(k+1+s) \end{aligned} \tag{9.19}$$

by replacing the time instant k by $k+1+s$. Also, from Eqn. 9.18 it follows that

$$\mathbf{x}(k-s) = \mathbf{\Gamma}_s^{\dagger} \left(\underline{\mathbf{y}}_s(k) - \mathbf{H}_s \underline{\mathbf{u}}_s(k) - \mathbf{G}_s \underline{\boldsymbol{\phi}}_s(k) + \mathbf{H}_s \underline{\mathbf{v}}_s(k) - \underline{\mathbf{o}}_s(k) \right) \tag{9.20}$$

where † stands for the Moore-Penrose pseudo inverse. Because the pair $(\mathbf{A}, \mathbf{C})$ preserves the assumed observability of $(\mathbf{A}_c, \mathbf{C}_c)$ in the original CT system, $\mathbf{\Gamma}_s$ is of full column rank.

On the other hand, performing repeated recursions on the first line of Eqn. 9.11 shows that

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}^{s+1} \mathbf{x}(k-s) + \left[\mathbf{A}^s \mathbf{B} \ \cdots \ \mathbf{B}\right] \left(\underline{\mathbf{u}}_s(k) - \underline{\mathbf{v}}_s(k)\right) + \left[\mathbf{A}^s \mathbf{E} \ \cdots \ \mathbf{E}\right] \underline{\boldsymbol{\phi}}_s(k) \\ &= \mathbf{L}_p \underline{\mathbf{p}}_s(k) + \mathbf{L}_\omega \underline{\boldsymbol{\omega}}_s(k) \end{aligned} \tag{9.21}$$

where Eqn. 9.20 has been employed. In addition, $\underline{\mathbf{p}}_s(k) = \left[\underline{\mathbf{y}}_s'(k) \ \ \underline{\mathbf{u}}_s'(k) \right]' \in \Re^{pm_s + gl_s}$,

$$\mathbf{L}_p = \left[\mathbf{A}^{s+1} \mathbf{\Gamma}_s^{\dagger} \ \ \left(\left[\mathbf{A}^s \mathbf{B} \ \cdots \ \mathbf{A} \mathbf{B} \ \ \mathbf{B}\right] - \mathbf{A}^{s+1} \mathbf{\Gamma}_s^{\dagger} \mathbf{H}_s\right)\right] \in \Re^{n \times (pm_s + gl_s)},$$

$$\mathbf{L}_\omega = \left[-\mathbf{L}_p \quad \left([\mathbf{A}^s\,\mathbf{E} \cdots \mathbf{A}\,\mathbf{E}\;\; \mathbf{E}] - \mathbf{A}^{s+1}\,\boldsymbol{\Gamma}_s^\dagger\,\mathbf{G}_s\right)\right] \in \Re^{n\times(pm_s+gl_s+gn_s)},$$

and $\underline{\boldsymbol{\omega}}_s(k) = \left[\begin{array}{ccc} \underline{\mathbf{o}}_s'(k) & \underline{\mathbf{v}}_s'(k) & \underline{\boldsymbol{\phi}}_s'(k) \end{array}\right]' \in \Re^{pm_s+gl_s+gn_s}$.

The incorporation of Eqn. 9.21 into Eqn. 9.19 yields

$$\begin{aligned} \underline{\mathbf{y}}_s(k+1+s) - \mathbf{H}_s\underline{\mathbf{u}}_s(k+1+s) &= \boldsymbol{\Gamma}_s\mathbf{L}_p\underline{\mathbf{p}}_s(k) + \boldsymbol{\Gamma}_s\mathbf{L}_\omega\underline{\boldsymbol{\omega}}_s(k) \\ &\quad + \mathbf{L}_\omega^f\underline{\boldsymbol{\omega}}_s(k+1+s) \end{aligned} \tag{9.22}$$

where, $\mathbf{L}_\omega^f = [\mathbf{I}_{m_s+p_s} \mid -\mathbf{H}_s \mid \mathbf{G}_s]$, and $\underline{\boldsymbol{\omega}}_s(k+1+s)$ is structurally similar to $\underline{\boldsymbol{\omega}}_s(k)$. Eqn. 9.22 is a sample-wise stacked equation, which lays the foundation for the following derivation of the subspace algorithms.

The Block-wise Stacked Equation

The following block Hankel matrix for the inputs can be constructed,

$$\underline{\mathbf{U}}_{k,s,N} = \left[\begin{array}{cccc} \underline{\mathbf{u}}(k) & \underline{\mathbf{u}}(k+1) & \cdots & \underline{\mathbf{u}}(k+N-1) \\ \underline{\mathbf{u}}(k+1) & \underline{\mathbf{u}}(k+2) & \cdots & \underline{\mathbf{u}}(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ \underline{\mathbf{u}}(k+s) & \underline{\mathbf{u}}(k+s+1) & \cdots & \underline{\mathbf{u}}(k+s+N-1) \end{array}\right] \in \Re^{gl_s\times N},$$

where the first subscript of $\underline{\mathbf{U}}_{k,s,N}$ indicates the time stamp of the $(1,1)$ block element of the matrix, and N is a large positive integer tending to ∞. The output and noise Hankel matrices have similar formats, which are denoted by $\underline{\mathbf{Y}}_{k,s,N} \in \Re^{pm_s\times N}$ and $\underline{\boldsymbol{\Omega}}_{k,s,N} \in \Re^{(pm_s+gl_s+gn_s)\times N}$ respectively.

Using the block Hankel data matrices, one can expand the sample-wise stacked equation (Eqn. 9.22) to the following block-wise stacked equation:

$$\underline{\mathbf{Y}}_{L+s+1,s,N} = \mathbf{H}_s\underline{\mathbf{U}}_{L+s+1,s,N} + \boldsymbol{\Gamma}_s\mathbf{L}_p\underline{\mathbf{P}}_{L,s,N} + \left[\boldsymbol{\Gamma}_s\mathbf{L}_\omega \quad \mathbf{L}_\omega^f\right]\left[\begin{array}{c} \underline{\boldsymbol{\Omega}}_{L,s,N} \\ \underline{\boldsymbol{\Omega}}_{L+s+1,s,N} \end{array}\right] \tag{9.23}$$

where $\underline{\mathbf{P}}_{L,s,N} = \left[\begin{array}{cc} \underline{\mathbf{Y}}_{L,s,N}' & \underline{\mathbf{U}}_{L,s,N}' \end{array}\right]' \in \Re^{(pm_s+gl_s)\times N}$.

Development of Identification Algorithms

In order to obtain consistent estimates of $\boldsymbol{\Gamma}_s$ and $\mathbf{H}_s$, we remove the noise-related terms in Eqn. 9.23. With a choice of $L = 2s+2$, post-multiplying Eqn. 9.23 by

matrix $[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}]$ generates

$$\begin{aligned}
&\frac{1}{N}\underline{\mathbf{Y}}_{3s+3,s,N}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}] \\
= \;& \frac{1}{N}\mathbf{\Gamma}_s\mathbf{L}_p\underline{\mathbf{P}}_{2s+2,s,N}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}] + \frac{1}{N}\mathbf{H}_s\underline{\mathbf{U}}_{3s+3,s,N}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}] \\
+ \;& \frac{1}{N}[\mathbf{\Gamma}_s\mathbf{L}_\omega \;\; \mathbf{L}^f_\omega]\begin{bmatrix}\underline{\mathbf{\Omega}}_{2s+2,s,N}\\ \underline{\mathbf{\Omega}}_{3s+3,s,N}\end{bmatrix}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}] \qquad (9.24)
\end{aligned}$$

For convenience, denote

$$\begin{aligned}
\underline{\mathbf{U}}_\circ &= \frac{1}{N}\underline{\mathbf{U}}_{3s+3,s,N}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}], \; \underline{\mathbf{Y}}_\circ = \frac{1}{N}\underline{\mathbf{Y}}_{3s+3,s,N}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}] \\
\underline{\mathbf{P}}_\circ &= \frac{1}{N}\underline{\mathbf{P}}_{2s+2,s,N}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}], \; \underline{\mathbf{E}}_\circ \\
&= \frac{1}{N}[\mathbf{\Gamma}_s\mathbf{L}_\omega \;\; \mathbf{L}^f_\omega]\begin{bmatrix}\underline{\mathbf{\Omega}}_{2s+2,s,N}\\ \underline{\mathbf{\Omega}}_{3s+3,s,N}\end{bmatrix}[\underline{\mathbf{P}}'_{0,s,N} \;\; \underline{\mathbf{P}}'_{s+1,s,N}].
\end{aligned}$$

Based on Eqn. 9.24, the least square (LS) estimate of $[\mathbf{\Gamma}_s\mathbf{L}_p \;\; \mathbf{H}_s]$ is

$$\begin{aligned}
[\hat{\mathbf{\Gamma}}_s\hat{\mathbf{L}}_p \;\; \hat{\mathbf{H}}_s] &= \underline{\mathbf{Y}}_\circ[\underline{\mathbf{P}}'_\circ \;\; \underline{\mathbf{U}}'_\circ]\left(\begin{bmatrix}\underline{\mathbf{P}}_\circ\\ \underline{\mathbf{U}}_\circ\end{bmatrix}[\underline{\mathbf{P}}'_\circ \;\; \underline{\mathbf{U}}'_\circ]\right)^{-1} \\
&= \left([\mathbf{\Gamma}_s\mathbf{L}_p \;\; \mathbf{H}_s]\begin{bmatrix}\underline{\mathbf{P}}_\circ\\ \underline{\mathbf{U}}_\circ\end{bmatrix} + \underline{\mathbf{E}}_\circ\right)[\underline{\mathbf{P}}'_\circ \;\; \underline{\mathbf{U}}'_\circ]\left(\begin{bmatrix}\underline{\mathbf{P}}_\circ\\ \underline{\mathbf{U}}_\circ\end{bmatrix}[\underline{\mathbf{P}}'_\circ \;\; \underline{\mathbf{U}}'_\circ]\right)^{-1} \qquad (9.25)
\end{aligned}$$

where, as in Chou and Verhaegen ([18]), one can show that $\underline{\mathbf{E}}_\circ$ vanishes asymptotically. This is so because $\underline{\mathbf{v}}(k)$, $\underline{\mathbf{o}}(k)$, and $\underline{\boldsymbol{\phi}}(k)$ are white noise vectors. Hence, if

$$\left(\begin{bmatrix}\underline{\mathbf{P}}_\circ\\ \underline{\mathbf{U}}_\circ\end{bmatrix}[\underline{\mathbf{P}}'_\circ \;\; \underline{\mathbf{U}}'_\circ]\right)^{-1}$$

exists, then $[\hat{\mathbf{\Gamma}}_s\hat{\mathbf{L}}_p \;\; \hat{\mathbf{H}}_s] \to [\mathbf{\Gamma}_s\mathbf{L}_p \;\; \mathbf{H}_s]$ as $N \to \infty$.

The reasons for the choice of $L = 2s+2$ are as follows. First, the minimum length of L is equal to $2s+2$. If $L < 2s+2$, it can be readily proved that $\underline{\mathbf{E}}_\circ \neq \mathbf{0}$ as $N \to \infty$. However, choosing a larger L can increase the probability that $\begin{bmatrix}\underline{\mathbf{P}}_\circ\\ \underline{\mathbf{U}}_\circ\end{bmatrix}[\underline{\mathbf{P}}'_\circ \;\; \underline{\mathbf{U}}'_\circ]$ is not invertible, as pointed out by Chou and Vergaegen ([18]).

To compute $[\mathbf{\Gamma}_s\mathbf{L}_p \quad \mathbf{H}_s]$, one need an algorithm that is more efficient and reliable than Eqn. 9.25. Thus, the following orthogonal-triangular decomposition is performed:

$$\begin{bmatrix} \underline{\mathbf{P}}_0 \\ \underline{\mathbf{Y}}_0 \\ \underline{\mathbf{U}}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & \mathbf{0} \\ \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} & \mathbf{R}_{44} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \end{bmatrix} \tag{9.26}$$

Subsequently,

$$\mathbf{H}_s = [\mathbf{R}_{33}\ \mathbf{0}]\,[\mathbf{R}_{43}\ \mathbf{R}_{44}]^{\dagger} \tag{9.27}$$

can be obtained. The detailed derivation of Eqn. 9.27 is documented in Appendix 9.C.

Furthermore,
performing a singular value decomposition (SVD) on $[\mathbf{R}_{31}\ \mathbf{R}_{32}]\,[\mathbf{R}_{41}\ \mathbf{R}_{42}]^{\perp}$ gives $[\mathbf{R}_{31}\ \mathbf{R}_{32}]\,[\mathbf{R}_{41}\ \mathbf{R}_{42}]^{\perp'} = \mathbf{U}_l\mathbf{\Lambda}\mathbf{V}_r'$. Consequently, we can select the first n vectors of $\mathbf{U}_l$ as the consistent estimate of $\mathbf{\Gamma}_s$ (up to a column space), i.e. $\mathbf{\Gamma}_s = \mathbf{U}_l(:, 1:n)$. The detailed derivation of the algorithm to compute $\mathbf{\Gamma}_s$ is presented in Appendix 9.D.

9.4.3 Optimal Design of $\mathbf{W}_o$

For the sake of completeness, the optimal design of $\mathbf{W}_o$ is discussed now. Eqn. 9.17 clearly shows that in a PRV the fault-contributed term is

$$\underline{\mathbf{e}}_o^f(k) = \mathbf{W}_o\underline{\mathbf{f}}_{s,y}(k) - \mathbf{W}_o\mathbf{H}_s\underline{\mathbf{f}}_{s,u}(k).$$

Calculation of $\mathbf{W}_o$

The PRV should have maximized sensitivity to the stacked fault vectors, $\underline{\mathbf{f}}_{s,y}(k)$ and $\underline{\mathbf{f}}_{s,u}(k)$. Geometrically, this means that besides satisfying $\mathbf{W}_o\mathbf{\Gamma}_s = \mathbf{0}$, $\mathbf{W}_o$ should have maximized covariance with matrix $\mathbf{H}_s$. In accordance with the work of Golub ([42], cf. pp 319-320) and that of Rao ([107], cf. pp 331-332),

$$\mathbf{W}_o' = \text{the eigenvectors of } \mathbf{\Gamma}_s^{\perp}\mathbf{H}_s\mathbf{H}_s' \text{ corresponding to non-zero eigenvalues} \tag{9.28}$$

Conditions for the Existence of a Non-trivial $\mathbf{W}_o$

From its definition, it is evident that $\mathbf{\Gamma}_s$ is a $pm_s \times n$ matrix and has rank n. Accordingly, the rank of the *left null space* of $\mathbf{\Gamma}_s$ is $pm_s - n$. Because $\mathbf{W}_o$ is located in such a null space, it has pm_s columns and $pm_s - n = pms + pm - n$ independent rows. Due to the choice of $s = n$ and $pm > 1$, $pms - n + pm > 0$ always holds. Consequently, a non-trivial solution to $\mathbf{W}_o$ is always available. We define a new convention, $\text{Rank}(\mathbf{W}_o) \equiv pms + pm - n$, which is the dimension of the PRV.

9.5 An Experimental Case Study

In this section, an experimental case study is conducted to test the effectiveness of the proposed scheme. We will demonstrate the detailed procedure for (1) identifying a residual model using NUSM data collected from an experimental pilot plant; (2) using the identified residual model to generate a PRV for fault detection; and (3) designing a set of structured residual vectors for fault isolation.

9.5.1 The Experimental Pilot Plant

The experimental pilot plant is a continuous stirred tank heater system (CSTHS) located in the Computer Process Control Laboratory at the University of Alberta. As shown in Figure 9.3, the CSTHS has two inputs, the cold water and the hot water, which are well mixed. The ultimate purpose of the CSTHS is to control the level and temperature of the water, both of which are also chosen to be the outputs.

9.5.2 Preliminary Work for FDD

Identification of the Residual Model for Fault Detection

From this pilot plant, a set of training data with 1920 points and a frame period of $T = 5$ secs was collected to identify the residual model. Within each frame period $[kT, kT+T]$ for $k = 0, 1, 2, \cdots$, the two inputs are sampled at instants kT and $kT+2$, while the two outputs are sampled at instants kT and $kT+3$. Thus, the lifted input and output vectors are

$$\underline{\mathbf{u}}(k) = \begin{bmatrix} \mathbf{u}(kT) \\ \mathbf{u}(kT+2) \end{bmatrix}, \ \underline{\mathbf{y}}(k) = \begin{bmatrix} \mathbf{y}(kT) \\ \mathbf{y}(kT+3) \end{bmatrix}.$$

Figure 9.3: Physical layout of the CSTHS system with the associated hardware

Based on the principles of mass and energy balances, the dynamics of the pilot plant can be represented by a second order system, i.e. the order of the model is $n = 2$. This resulted in two identified matrices: $\mathbf{\Gamma}_s \in \Re^{12\times 2}$, $\mathbf{H}_s \in \Re^{12\times 12}$, followed by a calculated matrix: $\mathbf{W}_o \in \Re^{10\times 12}$, where a value of 2 is selected for s. Further, with $\mathbf{\Gamma}_s$, $\mathbf{H}_s$, and $\mathbf{W}_o$, the residual model $\mathbf{M}_s = [\mathbf{W}_o | -\mathbf{W}_o\mathbf{H}_s] \in \Re^{10\times 24}$ is constructed. Furthermore, from the training data and $\mathbf{M}_s$, a sequence of PRVs are generated, from which the covariance matrix $\underline{\mathbf{R}}_{s,e} \in \Re^{10\times 10}$ is estimated.

Validation of the Identified Residual Model

In addition to the data used for identification, another sequence of data for the fault-free case was generated for model validation. From this data sequence and the identified residual model, i.e. $\mathbf{M}_s$, a sequence of PRVs, $\{\underline{\mathbf{e}}_s(k)\}$, is generated and depicted in Figure 9.4.

One can calculate $\mathrm{FD}(k) = \underline{\mathbf{e}}_s'(k)\underline{\mathbf{R}}_{s,e}^{-1}\underline{\mathbf{e}}_s(k)$ as the fault detection index (an estimate of the covariance matrix $\underline{\mathbf{R}}_{s,e}$ was given earlier). Previously mentioned, in the fault-free case, because $\underline{\mathbf{e}}_s(k) \sim \mathcal{N}\left(\mathbf{0}, \underline{\mathbf{R}}_{s,e}\right)$, $\mathrm{FD}(k)$ follows a chi-square distribution with degrees of freedom 10 ([58]). Therefore, at a level of significance $\alpha = 0.01$, the confidence limit for $\mathrm{FD}(k)$ is $\chi^2_{0.01}(10) = 23.209$.

The calculated $\{\mathrm{FD}(k)\}$ are plotted in Figure 9.5. Clearly, $\{\mathrm{FD}(k)\}$ is within its confidence limit, indicating that the identified residual model is valid.

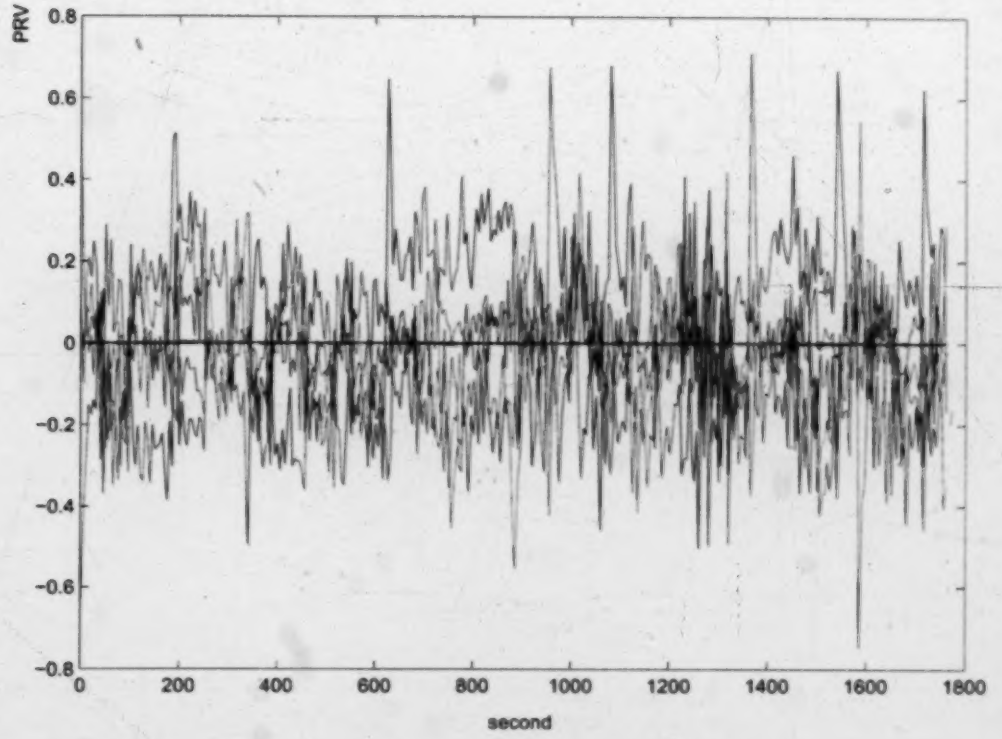

Figure 9.4: A sequence of PRVs generated from the validation data

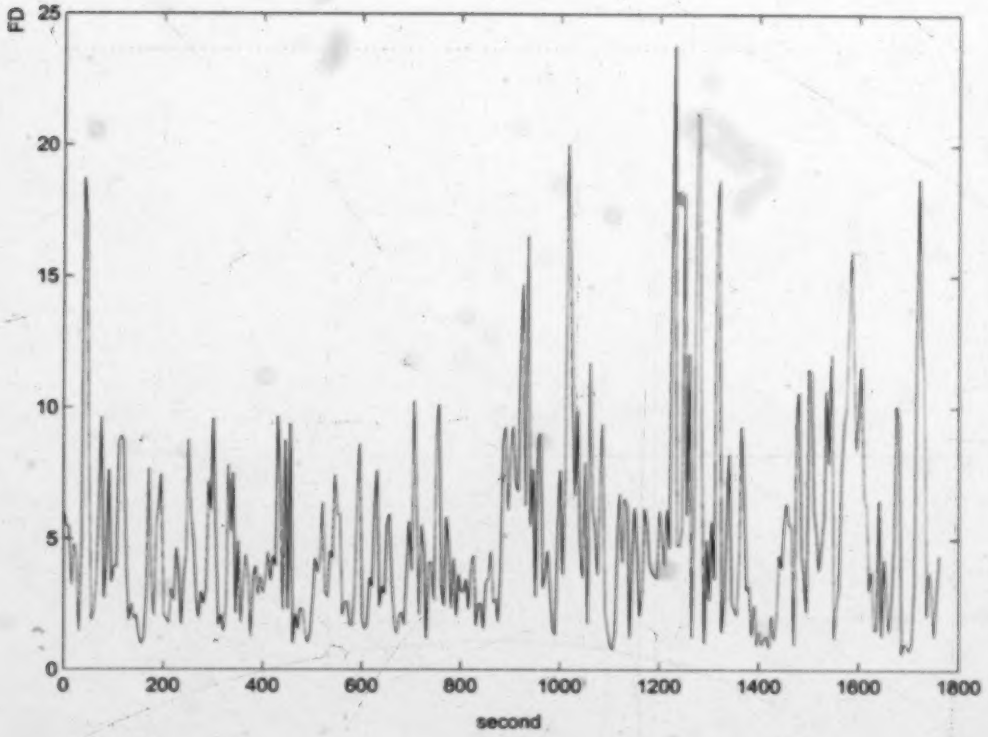

Figure 9.5: The fault detection index generated from the validation data. The dotted line represents the threshold, 23.209, for the index.

Table 9.1: Sensitivity and insensitivity of the 4 SRVs with respect to faulty sensors in the CSTHS system

	1st output sensor	2nd output sensor	1st input sensor	2nd input sensor
1st SRV	0	1	1	1
2nd SRV	1	0	1	1
3rd SRV	1	1	0	1
4th SRV	1	1	1	0

Calculation of the Residual Models for Fault Isolation

While we can use the PRV to conduct fault detection, we must transform the PRV into a set of SRVs to isolate each faulty sensor. Assume that at each time, only one sensor is faulty. Because 4 sensors (2 for the inputs, and 2 for the outputs) were installed in the CSTHS system, accordingly we must design 4 SRVs, each of which is insensitive to one faulty sensor but has maximized sensitivity to other faulty sensors.

More specifically, we design the i^{th} SRV to be insensitive to any fault in the i^{th} sensor, but to have maximized sensitivity to faults in other sensors for $i \in [1, ..., 4]$. This is termed *fault isolation logic*. The sensitivity and insensitivity of the 4 SRVs for the faulty sensors are summarized in Table 9.1. In the table, a '0'/'1' means the insensitivity/sensitivity of a SRV to a faulty sensor.

Mathematically, the i^{th} SRV is

$$\underline{\mathbf{r}}_{s,i}(k) = \mathbf{W}_i \underline{\mathbf{e}}_s(k) = \mathbf{W}_i \mathbf{M}_s \begin{bmatrix} \underline{\mathbf{y}}_s(k) \\ \underline{\mathbf{u}}_s(k) \end{bmatrix}.$$

Therefore, the model for the i^{th} SRV is $\mathbf{W}_i\mathbf{M}_s$. With known $\mathbf{M}_s$ and the isolation logic illustrated in Table 9.1, using algorithms similar to those presented by Li and Shah ([70]), we can obtain 4 transformation matrices $\mathbf{W}_i \in \Re^{4\times 10}$, and consequently the structured residual models, $\mathbf{W}_i\mathbf{M}_s \in \Re^{4\times 24}$, $\forall\, i \in [1, ..., 4]$.

When designing the 4 SRVs for isolation, a question arises: What are the conditions to guarantee their existence? Similar to the analysis in [70], we can provide an answer to this question as follows.

Because the model for each SRV is $\mathbf{W}_i\mathbf{M}_s$, the existence of a SRV depends entirely

on the existence of a non-trivial matrix $\mathbf{W}_i$, given $\mathbf{M}_s$. For each output sensor fault, $p(s+1)$ elements in $\underline{\mathbf{y}}_s(k)$ will be affected by the fault. Similarly, each input sensor fault will affect $q(s+1)$ elements in $\underline{\mathbf{u}}_s(k)$. To make a SRV insensitive to one fault, one must design a $\mathbf{W}_i$ such that it is orthogonal to the fault-related $p(s+1)$ or $q(s+1)$ columns in $\mathbf{M}_s$. Keep in mind that the rank of $\mathbf{M}_s$ is $\text{Rank}(\mathbf{W}_o)$. Assuming that the aforementioned $p(s+1)$ or $q(s+1)$ columns have a rank $p(s+1)$ or $q(s+1)$, it can be inferred that $\mathbf{W}_i$ has at least $\text{Rank}(\mathbf{W}_o) - p(s+1)$ or $\text{Rank}(\mathbf{W}_o) - q(s+1)$ independent rows. If the number of independent rows in $\mathbf{W}_i$ is equal to or larger than 1, the existence of a SRV is confirmed.

In this case study, we have $\text{Rank}(\mathbf{W}_o) = 10$, $p(s+1) = 2(2+1) = 6$, $q(s+1) = 2(2+1) = 6$. Therefore, each $\mathbf{W}_i$ has $\text{Rank}(\mathbf{W}_o) - p(s+1) = \text{Rank}(\mathbf{W}_o) - q(s+1) = 4$ independent rows, $\forall\, i \in [1, ..., 4]$, indicating the existence of the 4 SRVs.

Decision Making for FDD

For better visualization, we scale $\text{FD}(k)$ to have unit confidence limit, resulting in a scaled fault detection index, $\bar{\text{FD}}(k) = \text{FD}(k)/23.209$. Consequently, for real-time sampled data, while $\bar{\text{FD}}(k) < 1$ indicates the fault-free case, $\bar{\text{FD}}(k) \geq 1$ triggers alarms for any faults in sensor(s) ([9]).

After fault detection, we can isolate the faulty sensor as follows. Because $\underline{\mathbf{r}}_{s,i}(k)$ is a linear combination of $\underline{\mathbf{e}}_s(k)$, in the fault-free case it can be inferred that $\underline{\mathbf{r}}_{s,i}(k) \sim (\mathbf{0}, \mathbf{W}_i \underline{\mathbf{R}}_{s,e} \mathbf{W}_i')$ ([58]). In addition, because $\underline{\mathbf{r}}_{s,i}(k)$ is insensitive to a fault in the i^{th} sensor, $\underline{\mathbf{r}}_{s,i}(k) \sim (\mathbf{0}, \mathbf{W}_i \underline{\mathbf{R}}_{s,e} \mathbf{W}_i')$ when the i^{th} sensor is faulty but the other sensors are fault-free.

We define a fault isolation index $\text{FI}_i(k) = \underline{\mathbf{r}}_{s,i}(k) \underline{\mathbf{R}}_{s,i}^{-1} \underline{\mathbf{r}}_{s,i}(k)$, where $\underline{\mathbf{R}}_{s,i}^{-1} = \mathbf{W}_i \underline{\mathbf{R}}_{s,e} \mathbf{W}_i'$. Then in the afore-mentioned two cases, $\text{FI}_i(k) \sim \chi^2(4)$ ([58]). With $\alpha = 0.01$, the confidence limit for $\text{FI}_i(k)$ is 13.277. Similarly, we scale the fault isolation indices to have unit confidence limit, obtaining the scaled fault detection indices, $\bar{\text{FI}}_i(k) = \text{FI}_i(k)/13.277$.

After fault detection, if $\bar{\text{FI}}_i(k) < 1$ but $\bar{\text{FI}}_j(k) \geq 1$ for $\{i, j \in [1, 4]\}$ and $\{i \neq j\}$, it can be decided that the i^{th} sensor is faulty. It must be noted that in the CSTHS system the i^{th} sensor refers to the i^{th} output sensor if $i \in [1, 2]$, or the $(i-2)^{th}$ input sensor if $i \in [3, 4]$.

9.5.3 FDD Results

Although we have done FDD for many cases in the course of this study, FDD results for only two are presented here. In each case, a fault is simulated by introducing a drift or a noise to one of the 4 sensors at a time. Furthermore, note that in each case, a sequence of test data from $t = 0$ to $t = 1750$ seconds is sampled in the same manner as the training data.

Case 1

A drift fault simulated by $0.001(t-t_f)$ is introduced to one sensor at $t_f = 980$ seconds. The FDD results are depicted in Figure 9.6. FD is the scaled fault detection index, and $\{FI_1, FI_2, FI_3, FI_4\}$ are the scaled fault isolation indices (the same conventions will be used in the second case). It is evident in Figure 9.6 that FD is beyond the unit confidence limit after the occurrence of the fault. Therefore, fault detection has been successfully achieved. FI_1 is unaffected by the fault, i.e. it is below the confidence limit (at few periods of time it is beyond the limit), while $\{FI_2, FI_3, FI_4\}$ are affected, i.e. they are beyond the confidence limit. The sensitivity of the 4 fault isolation indices can be characterized by a binary code [0 1 1 1]. In accordance with the isolation logic described earlier, it can be inferred that the first output sensor has a fault. It must be pointed out that there is a delay in detecting and isolating the fault. This is so because the drift fault is an incipient fault that evolves with time very slowly.

Case 2

In this case, a zero mean Gaussian distributed noise with variance equal to 0.2^2 is introduced to a sensor at $t_f = 780$ seconds. This simulates the fault of precision degradation. The FDD results are depicted in Figure 9.7. Because the sensitivity of the isolation indices to the fault is [1 0 1 1], it can be concluded that the second output sensor has a fault.

To quantify the sensitivity of the proposed FDD scheme, we define the fault-to-signal ratio as follows ,

$$r_{f/s} = \frac{\sum_{k=k_f}^{N_0} \|\underline{\mathbf{f}}_y(k)\|}{\sum_{k=k_f}^{N_0} \|\underline{\mathbf{y}}^*(k)\|}\%.$$

$\|\mathbf{f}_y(k)\|$ is the 2-norm of the lifted fault vector, $\|\underline{\mathbf{y}}^*(k)\|$ is the 2-norm of the fault-free lifted output vector at the k^{th} frame period, k_f is the frame period at which the fault

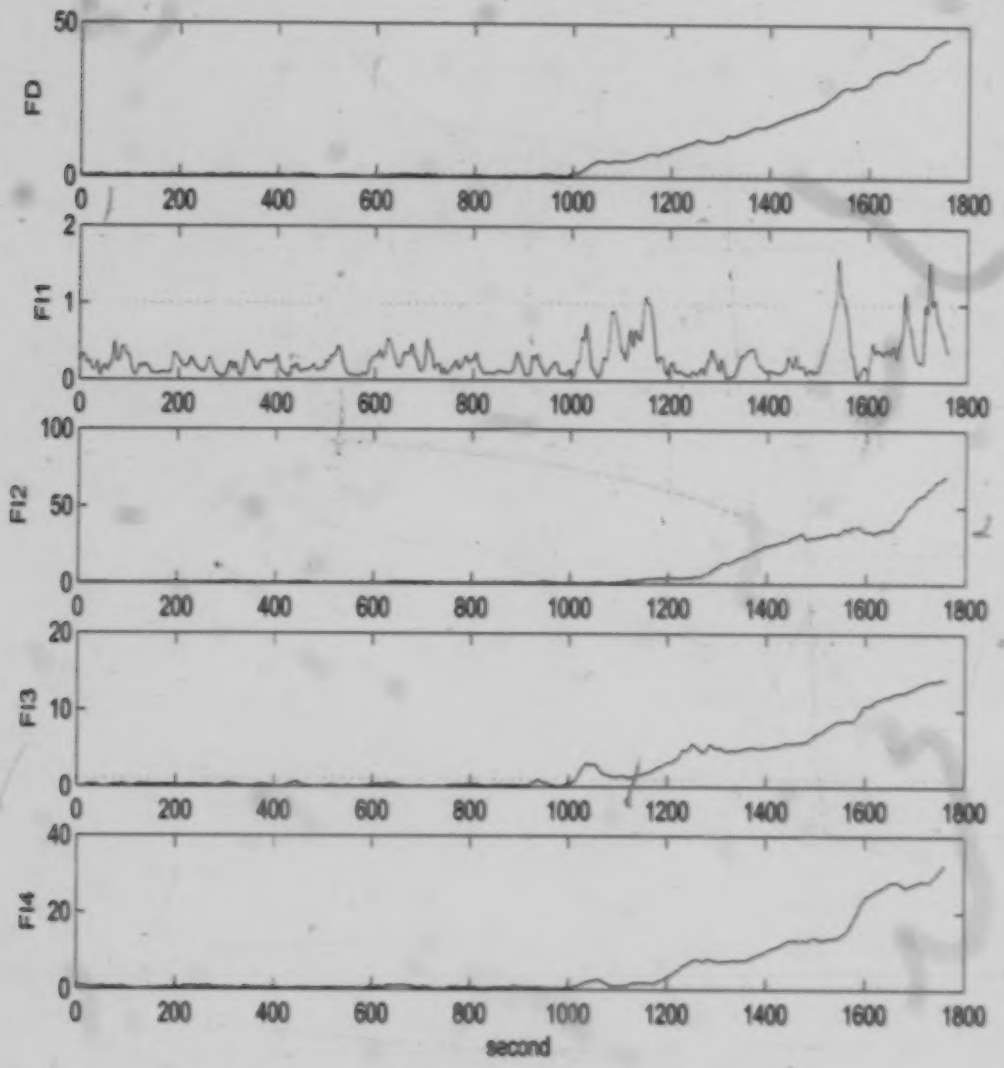

Figure 9.6: Detection and isolation of a fault in the 1st sensor. The sensitivity of the isolation indices to the fault is [0 1 1 1]. The dotted line in each subplot represents the threshold, 1, for the scaled detection and isolation indices.

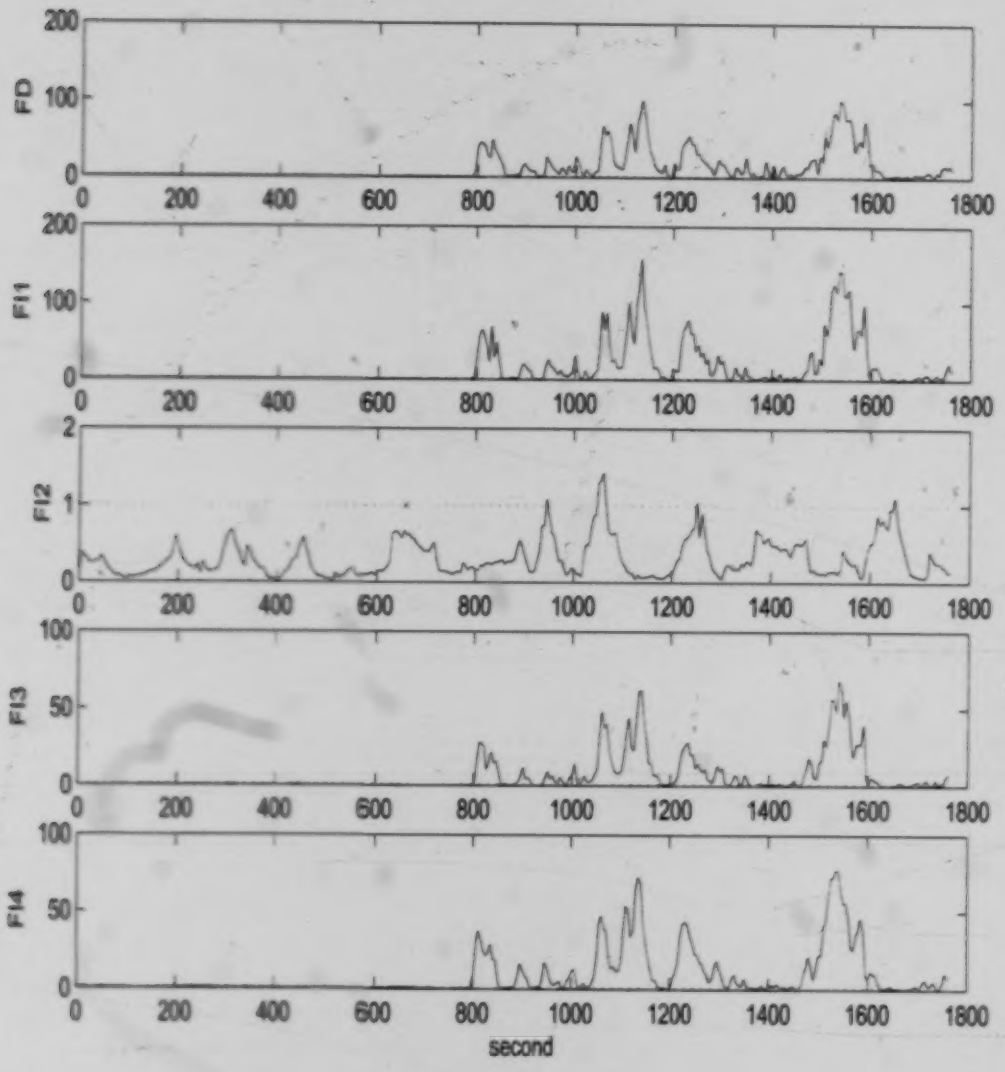

Figure 9.7: Detection and isolation of a fault in the 2^{nd} sensor. The sensitivity of the isolation indices to the fault is [1 0 1 1]. The dotted line in each subplot represents the threshold, 1, for the scaled detection and isolation indices.

occurs, and the N_0 is the number of frame periods in the test data. In this case, $k_f = t_f/5 = 780/5 = 156$, $N_0 = 1750/5 = 250$, and $r_{f/a} = 11.05\%$.

9.6 Concluding remarks

A novel subspace approach to identification of residual models for FDD with NUMS data has been proposed. In comparison to the uniformly sampled multirate FDD work represented in the literature, this new approach is more generic and applicable to a wider class of processes. Using the identified model, a PRV is generated for fault detection by extending the Chow-Willsky method. For the PRV, sensitivity to any faults has been maximized.

This approach has been applied to an experimental pilot plant, i.e. the CSTHS system in the Computer Process Control laboratory at The University of Alberta, where the model's validity was assessed. Various types of sensor faults in the CSTHS system, including drift and precision degradation, are successfully detected and isolated. Therefore, the practicality and utility of the proposed methodology have been demonstrated.

9.A Derivation of Eqn. 9.6

Over the period $t \in [kT, kT+T]$,

$$\begin{aligned}\int_{kT}^{kT+T} e^{\mathbf{A}_c(kT+T-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t)+\boldsymbol{\phi}(t)\right]dt &= \int_{kT+t_1}^{kT+t_2} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}[\mathbf{B}_c\tilde{\mathbf{u}}(t)dt+\boldsymbol{\phi}(t)dt] \\ &+ \int_{kT+t_2}^{kT+t_3} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}[\mathbf{B}_c\tilde{\mathbf{u}}(t)dt+\boldsymbol{\phi}(t)dt]+\cdots \\ &+ \int_{kT+t_g}^{kT+t_{g+1}} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}[\mathbf{B}_c\tilde{\mathbf{u}}(t)dt+\boldsymbol{\phi}(t)dt]\end{aligned} \tag{9.A.1}$$

where $t_1 = 0$ and $t_{g+1} = T$.

Assume that $\tilde{\mathbf{u}}(t)$ and $\boldsymbol{\phi}(t)$ are *piece-wise constant* within the interval $[kT+t_i, kT+t_{i+1}]$

for $i = [1, ..., g]$. Eqn. 9.A.1 can be easily evaluated as

$$\begin{aligned}\int_{kT}^{kT+T} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t)+\boldsymbol{\phi}(t)\right]dt = \\ \int_{kT+t_1}^{kT+t_2} \mathbf{e}^{\mathbf{A}(_ckT+T-t)}\mathbf{B}_c dt\left[\tilde{\mathbf{u}}(kT+t_1)+\boldsymbol{\phi}(kT+t_1)\right] \\ + \int_{kT+t_2}^{kT+t_3} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}\mathbf{B}_c dt\left[\tilde{\mathbf{u}}(kT+t_2)+\boldsymbol{\phi}(kT+t_2)\right]+\cdots \\ + \int_{kT+t_g}^{kT+t_{g+1}} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}\mathbf{B}_c dt\left[\tilde{\mathbf{u}}(kT+t_g)+\boldsymbol{\phi}(kT+t_g)\right]\end{aligned} \tag{9.A.2}$$

Denote

$$\begin{aligned}\mathbf{B}_i &\equiv \int_{kT+t_i}^{kT+t_{i+1}} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}\mathbf{B}_c dt = \int_{T-t_{i+1}}^{T-t_i} \mathbf{e}^{\mathbf{A}_c t}\mathbf{B}_c dt \\ \mathbf{E}_i &\equiv \int_{kT+t_i}^{kT+t_{i+1}} \mathbf{e}^{\mathbf{A}_c(kT+T-t)}dt = \int_{T-t_{i+1}}^{T-t_i} \mathbf{e}^{\mathbf{A}_c t}dt.\end{aligned} \tag{9.A.3}$$

Consequently, Eqn. 9.A.2 is equal to

$$\begin{aligned}\int_{kT}^{kT+T} \mathbf{e}^{-\mathbf{A}_c(kT+T-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t)+\boldsymbol{\phi}(t)\right]dt &= [\mathbf{B}_1\ \mathbf{B}_2\ \cdots\ \mathbf{B}_g]\underline{\tilde{\mathbf{u}}}(k)+[\mathbf{E}_1\ \mathbf{E}_2\ \cdots\ \mathbf{E}_g]\underline{\boldsymbol{\phi}}(k) \\ &= \mathbf{B}\ \underline{\tilde{\mathbf{u}}}(k)+\mathbf{E}\ \underline{\boldsymbol{\phi}}(k)\end{aligned} \tag{9.A.4}$$

where, $\mathbf{B} = [\mathbf{B}_1\ \mathbf{B}_2\ \cdots\ \mathbf{B}_g]$, $\mathbf{E} = [\mathbf{E}_1\ \mathbf{E}_2\ \cdots\ \mathbf{E}_g]$, and the definitions of $\underline{\tilde{\mathbf{u}}}(k)$ and $\underline{\boldsymbol{\phi}}(k)$ have been employed. Furthermore, using Eqn. 9.A.4 in Eqn. 9.5 gives Eqn. 9.6 immediately.

9.B Derivation of Eqn. 9.10

Evaluating Eqn. 9.8 at $\tau = t_i^j$, where $i = [1, ..., g]$ and for each i, $j = [1, ..., n_i]$, it can be seen that

$$\tilde{\mathbf{y}}(kT+t_i^j) = \mathbf{C}_c\mathbf{e}^{\mathbf{A}_c t_i^j}\mathbf{x}(k)+\mathbf{C}_c\int_{kT}^{kT+t_i^j} \mathbf{e}^{\mathbf{A}_c(kT+t_i^j-t)}\left[\mathbf{B}_c\tilde{\mathbf{u}}(t)+\boldsymbol{\phi}(t)\right]dt+\mathbf{D}_c\tilde{\mathbf{u}}(kT+t_i^j) \tag{9.B.5}$$

Note that $\tilde{\mathbf{u}}(kT+t_i^j) = \tilde{\mathbf{u}}(kT+t_i)$.

Since

$$\begin{aligned}
&\mathbf{C}_c \int_{kT}^{kT+t_i^j} \mathbf{e}^{\mathbf{A}_c(kT+t_i^j-t)}\mathbf{B}_c\tilde{\mathbf{u}}(t)\,dt \\
&\quad = \mathbf{C}_c \int_{kT}^{kT+t_2} \mathbf{e}^{\mathbf{A}_c(kT+t_i^j-t)}\mathbf{B}_c dt\ \tilde{\mathbf{u}}(kT) \\
&\qquad + \mathbf{C}_c \int_{kT+t_2}^{kT+t_3} \mathbf{e}^{\mathbf{A}_c(kT+t_i^j-t)}\mathbf{B}_c dt\ \tilde{\mathbf{u}}(kT+t_2) + \cdots \\
&\qquad + \mathbf{C}_c \int_{kT+t_i}^{kT+t_i^j} \mathbf{e}^{\mathbf{A}_c(kT+t_i^j-t)}\mathbf{B}_c dt\ \tilde{\mathbf{u}}(kT+t_i) \\
&\quad = \left(\mathbf{C}_c \int_{t_i^j-t_2}^{t_i^j} \mathbf{e}^{\mathbf{A}_c t}\mathbf{B}_c dt\right)\ \tilde{\mathbf{u}}(kT) + \left(\mathbf{C}_c \int_{t_i^j-t_3}^{t_i^j-t_2} \mathbf{e}^{\mathbf{A}_c t}\mathbf{B}_c dt\right)\ \tilde{\mathbf{u}}(kT+t_2) \\
&\qquad + \cdots \left(\mathbf{C}_c \int_{0}^{t_i^j-t_i} \mathbf{e}^{\mathbf{A}_c t}\mathbf{B}_c dt\right)\ \tilde{\mathbf{u}}(kT+t_i) \qquad (9.B.6)
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{C}_c \int_{kT}^{kT+t_i^j} \mathbf{e}^{\mathbf{A}_c(kT+t_i^j-t)}\boldsymbol{\phi}(t)\,dt \;&=\; \left(\mathbf{C}_c \int_{t_i^j-t_2}^{t_i^j} \mathbf{e}^{\mathbf{A}_c t} dt\right)\ \boldsymbol{\phi}(kT) \\
&+ \left(\mathbf{C}_c \int_{t_i^j-t_3}^{t_i^j-t_2} \mathbf{e}^{\mathbf{A}_c t} dt\right)\ \boldsymbol{\phi}(kT+t_2) \\
&+ \cdots \left(\mathbf{C}_c \int_{0}^{t_i^j-t_i} \mathbf{e}^{\mathbf{A}_c t} dt\right)\ \boldsymbol{\phi}(kT+t_i) \qquad (9.B.7)
\end{aligned}$$

where $\tilde{\mathbf{u}}(t) = \tilde{\mathbf{u}}(kT+t_i)$ and $\boldsymbol{\phi}(t) = \boldsymbol{\phi}(kT+t_i)$ for $t \in [kT+t_i, kT+t_i^j]$, the combination of Eqns. 9.B.5-9.B.7 show

$$\begin{aligned}
\begin{bmatrix} \tilde{\mathbf{y}}(kT+t_i^1) \\ \tilde{\mathbf{y}}(kT+t_i^2) \\ \vdots \\ \tilde{\mathbf{y}}(kT+t_i^{n_i}) \end{bmatrix} \;&=\; \mathbf{C}_i\mathbf{x}(k) + [\mathbf{D}_{i,1}\ \cdots\ \mathbf{D}_{i,i}] \begin{bmatrix} \tilde{\mathbf{u}}(kT+t_1) \\ \tilde{\mathbf{u}}(kT+t_2) \\ \vdots \\ \tilde{\mathbf{u}}(kT+t_i) \end{bmatrix} \\
&+ [\mathbf{J}_{i,1}\ \cdots\ \mathbf{J}_{i,i}] \begin{bmatrix} \boldsymbol{\phi}(kT+t_1) \\ \boldsymbol{\phi}(kT+t_2) \\ \vdots \\ \boldsymbol{\phi}(kT+t_i) \end{bmatrix} \qquad (9.B.8)
\end{aligned}$$

where

$$\mathbf{C}_i = \begin{bmatrix} \mathbf{C}_c \mathbf{e}^{\mathbf{A}_c t_i^1} \\ \mathbf{C}_c \mathbf{e}^{\mathbf{A}_c t_i^2} \\ \vdots \\ \mathbf{C}_c \mathbf{e}^{\mathbf{A}_c t_i^{n_i}} \end{bmatrix}, \quad \mathbf{D}_{i,i} = \begin{bmatrix} \mathbf{C}_c \int_0^{t_i^1} \mathbf{e}^{\mathbf{A}_c t} \mathbf{B}_c dt + \mathbf{D}_c \\ \mathbf{C}_c \int_0^{t_i^2} \mathbf{e}^{\mathbf{A}_c t} \mathbf{B}_c dt + \mathbf{D}_c \\ \vdots \\ \mathbf{C}_c \int_0^{t_i^{n_i}} \mathbf{e}^{\mathbf{A}_c t} \mathbf{B}_c dt + \mathbf{D}_c \end{bmatrix}, \quad \mathbf{J}_{i,i} = \begin{bmatrix} \mathbf{C}_c \int_0^{t_i^1} \mathbf{e}^{\mathbf{A}_c t} dt \\ \mathbf{C}_c \int_0^{t_i^2} \mathbf{e}^{\mathbf{A}_c t} dt \\ \vdots \\ \mathbf{C}_c \int_0^{t_i^{n_i}} \mathbf{e}^{\mathbf{A}_c t} dt \end{bmatrix},$$

and for $j = [1, ..., i-1]$,

$$\mathbf{D}_{i,j} = \begin{bmatrix} \mathbf{C}_c \int_{t_i^1 - t_{j+1}}^{t_i^1 - t_j} \mathbf{e}^{\mathbf{A}_c t} \mathbf{B}_c dt \\ \mathbf{C}_c \int_{t_i^2 - t_{j+1}}^{t_i^2 - t_j} \mathbf{e}^{\mathbf{A}_c t} \mathbf{B}_c dt \\ \vdots \\ \mathbf{C}_c \int_{t_i^{n_i} - t_{j+1}}^{t_i^{n_i} - t_j} \mathbf{e}^{\mathbf{A}_c t} \mathbf{B}_c dt \end{bmatrix}, \quad \mathbf{J}_{i,j} = \begin{bmatrix} \mathbf{C}_c \int_{t_i^1 - t_{j+1}}^{t_i^1 - t_j} \mathbf{e}^{\mathbf{A}_c t} dt \\ \mathbf{C}_c \int_{t_i^2 - t_{j+1}}^{t_i^2 - t_j} \mathbf{e}^{\mathbf{A}_c t} dt \\ \vdots \\ \mathbf{C}_c \int_{t_i^{n_i} - t_{j+1}}^{t_i^{n_i} - t_j} \mathbf{e}^{\mathbf{A}_c t} dt \end{bmatrix}.$$

Using Eqn. 9.B.8 in Eqn. 9.8 gives $\underline{\tilde{\mathbf{y}}}(k) = \mathbf{C}\,\mathbf{x}(k) + \mathbf{D}\,\underline{\tilde{\mathbf{u}}}(k) + \mathbf{J}\,\underline{\boldsymbol{\phi}}(k)$, where

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_g \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_{1,1} & \mathbf{0} & & \cdots & \mathbf{0} \\ \mathbf{D}_{2,1} & \mathbf{D}_{2,2} & \mathbf{0} & & \vdots \\ \vdots & & \ddots & & \\ \vdots & & & & \\ \mathbf{D}_{g,1} & \mathbf{D}_{g,2} & \mathbf{D}_{g,3} & \cdots & \mathbf{D}_{g,g} \end{bmatrix},$$

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{1,1} & \mathbf{0} & & \cdots & \mathbf{0} \\ \mathbf{J}_{2,1} & \mathbf{J}_{2,2} & \mathbf{0} & & \vdots \\ \vdots & & \ddots & & \\ \vdots & & & & \\ \mathbf{J}_{g,1} & \mathbf{J}_{g,2} & \mathbf{J}_{g,3} & \cdots & \mathbf{J}_{g,g} \end{bmatrix}.$$

9.C Derivation of Eqn. 9.27

The use of Eqn. 9.26 in Eqn. 9.24 yields

$$\mathbf{R}_{31}\mathbf{Q}_1 + \mathbf{R}_{32}\mathbf{Q}_2 + \mathbf{R}_{33}\mathbf{Q}_3 = \boldsymbol{\Gamma}_s \mathbf{L}_p \begin{bmatrix} \mathbf{R}_{11}\mathbf{Q}_1 \\ \mathbf{R}_{21}\mathbf{Q}_1 + \mathbf{R}_{22}\mathbf{Q}_2 \end{bmatrix} + \mathbf{H}_s \left(\mathbf{R}_{41}\mathbf{Q}_1 + \mathbf{R}_{42}\mathbf{Q}_2 + \mathbf{R}_{43}\mathbf{Q}_3 + \mathbf{R}_{44}\mathbf{Q}_4\right) \quad (9.C.9)$$

Note that $\mathbf{Q}_i\mathbf{Q}_j^T = \begin{cases} \mathbf{I}, & i=j \\ \mathbf{0}, & i \neq j \end{cases}$, with $\mathbf{I}$ representing an identity matrix of appropriate dimension.

Post-multiplying Eqn. 9.C.9 by $\mathbf{Q}_3^T \in \Re^{(2pm_s+2gl_s)\times pm_s}$ and $\mathbf{Q}_4^T \in \Re^{(2pm_s+2gl_s)\times gl_s}$, respectively, we arrive at $[\mathbf{R}_{33}\ \ \mathbf{0}] = \mathbf{H}_s[\mathbf{R}_{43}\ \ \mathbf{R}_{44}]$. Therefore, $\mathbf{H}_s = [\mathbf{R}_{33}\ \mathbf{0}]\,[\mathbf{R}_{43}\ \mathbf{R}_{44}]^\dagger$.

9.D Derivation of the algorithm to compute Γ_s

Post-multiplying Eqn. 9.C.9 by $\mathbf{Q}_1^T \in \Re^{(2pm_s+2gl_s)\times pm_s}$ and $\mathbf{Q}_2^T \in \Re^{(2pm_s+2gl_s)\times gl_s}$, respectively, produces

$$[\mathbf{R}_{31}\ \ \mathbf{R}_{32}] = \mathbf{\Gamma}_s\mathbf{L}_p \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix} + \mathbf{H}_s\,[\mathbf{R}_{41}\ \ \mathbf{R}_{42}].$$

Consequently,

$$[\mathbf{R}_{31}\ \ \mathbf{R}_{32}]\,[\mathbf{R}_{41}\ \ \mathbf{R}_{42}]^\perp = \mathbf{\Gamma}_s\mathbf{L}_p \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix} [\mathbf{R}_{41}\ \ \mathbf{R}_{42}]^\perp,$$

where $(\ )^\perp$ stands for the right null space of the argument, i.e. $[\mathbf{R}_{41}\ \mathbf{R}_{42}]\,[\mathbf{R}_{41}\ \ \mathbf{R}_{42}]^\perp = \mathbf{0}$.

Furthermore, performing a singular value decomposition (SVD) on $[\mathbf{R}_{31}\ \ \mathbf{R}_{32}]\,[\mathbf{R}_{41}\ \ \mathbf{R}_{42}]^\perp$ gives $[\mathbf{R}_{31}\ \ \mathbf{R}_{32}]\,[\mathbf{R}_{41}\ \ \mathbf{R}_{42}]^\perp = \mathbf{U}_l\mathbf{\Lambda}\mathbf{V}_r'$, assuming that

$$\mathbf{L}_p \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix} [\mathbf{R}_{41}\ \ \mathbf{R}_{42}]^\perp$$

be of rank n. Finally, we select the first n vectors of $\mathbf{U}_l$ as the consistent estimate of $\mathbf{\Gamma}_s$ (up to a column space).

10

Concluding Remarks and Future Work

10.1 Concluding Remarks

The main contributions of this thesis are:

- A tutorial overview of existing system identification approaches – including prediction error method, subspace identification method and multivariate statistical regression method – is provided.

- Existing fault detection and diagnosis approaches and current research directions are reviewed. The key components of FDD, i.e process model and redundant measurements, are summarized and explained in detail.

- The concepts of fault detectability and strong fault detectability are defined, and criteria of fault detectability and strong fault detectability are proved. A special strong fault detectability condition for output sensor faults is given in

an easily understood manner. The results are applied to an inverted pendulum example in order to illustrate the effectiveness of the conditions.

- A set of softsensors for a complex chemical process in an industrial environment is designed and implemented. The partial least squares (PLS) method is used in this application. Due to the complexity of the process, nonlinear transformation and input variable selection are considered – resulting in a significant improvement in softsensor performance. Finally, the developed softsensors are used by Millar Western with a satisfactory level of performance.

- A new canonical variate analysis (CVA) for ill-conditioned data is developed. Using Cayley-Hamilton theorem and Krylov space, the new CVA method can handle data with collinearity, while still preserving the orthogonality among latent variables in an optimal manner. The new CVA method is extended to dynamic case.

- A robust sensor/actuator fault detection and diagnosis method dealing with the case of process uncertainties (including disturbances and model plant mismatch) is proposed. This method can completely decouple the effects of uncertainties for sensor fault detection and diagnosis. For actuator fault detection and diagnosis, the method can eliminate the major components of process uncertainties.

- An optimization procedure is applied to solve the multiplicative fault detection and diagnosis. The techniques of data reconciliation and gross error detection are used as theoretical bases for the procedure.

- A fault detection and diagnosis method dealing with multirate data is proposed, wherein different variables are measured at different sampling rates. Starting with a continuous time process, one can obtain its discrete time counterpart by using a lifting technique under multirate situation.

The new techniques were evaluated in several real and simulated situations:

- The detectability criteria were evaluated by a well-known inverted pendulum system under simulated conditions.

- A systematic approach for softsensor development was applied to Millar Western Forest Products Ltd., a plant that uses a bleached chemi-thermomechanical pulp (BCTMP) process at Whitecourt, Alberta, Canada.

- Canonical variate analysis (CVA) for ill-conditioned data was evaluated with data from a pulp and paper mill for the static case and by a simulated continuously stirred tank reactor (CSTR) system for the dynamic case.
- The robust sensor/actuator fault detection and diagnosis technique was applied on both a simulated CSTR system and a laboratory continuously stirred tank heater (CSTH) system.
- Detection and diagnosis procedures for multiplicative faults were implemented on a chemical tank inventory system at Millar Western Forest Products Ltd., Whitecourt, Alberta, Canada.
- Multirate fault detection and diagnosis techniques were evaluated at a laboratory CSTH system.

10.2 Recommendations for Future Research

Problems that still need to be investigated include:

- Establish a set of criteria for fault isolability. As was achieved on fault detectability in Chapter 4, fault isolability should be a property of the system under consideration. In such a case, the criteria of fault isolability can be determined solely by the system matrices. The problem of fault isolability is essentially to determine under what conditions two different faults lie in different directions in the residual space.
- Analyze the effect of parity space order. From Chapter 4, it is known that the selection of s does not affect the detectability of output sensor fault provided $s \geq n - 1$. However, the selection of s may affect the detectability of other types of faults. Consequently, one can determine the "optimal" selection of parity space order based on the results.
- Compare parity space and observer-based methods. The parity space method and observer based method have identical starting points, i.e. the state space representation of a system. Consequently, the following questions arise: are there any differences between these two methods? What are the advantages and/or disadvantages of each method? Are the methods equivalent under some conditions?

- Explore detection and isolation of model-plant mismatch (MPM) in detail. As mentioned in Chapter 7, a high quality process model is crucial for accurate FDD. An index of model quality must be developed. Furthermore, it would be very useful to develop a method of isolating the major source of modelling error from a multivariate transfer function matrix.

Bibliography

[1] Adebiyi, O.A. and A.B. Corripio. Dynamic neural networks partial least squares (DNNPLS) identification of multivariable processes. *Computers and Chemical Engineering*, 27:143–155, 2003. 60

[2] Akaike, H. *Trends and Progress in System Identification*, chapter Modern development of statistical methods. Pergamon Press, Elmsford, N.Y., 1981. 20

[3] Almasy, G.A. and T. Sztano. Checking and correction of measurements on the basis of linear system model. *Problems of control and information theory*, 4:57–69, 1975. 34

[4] Åström, K.J. and B. Wittenmark. *Computer Controlled Systems: Theory and Design*. Prentice-Hall, Englewood Cliffs, New Jersey, 2nd edition, 1990. 13

[5] Bagajewicz, M.J. *Process Plant Instrumentation: Design and Upgrade*. CRC, 2000. 131

[6] Basseville, M. Sequential detection of abrupt changes in spectral characteristics of digital signals. *IEEE Trans. Inform. Theo.*, 29(5):709–724, 1983. 26

[7] Basseville, M. Detection of changes in signals and systems– a survey. *Automatica*, 24(3):309–326, 1988. 26

[8] Basseville, M. On-board component fault detection and isolation using the statistical local approach. *Automatica*, 34(11):1391–1415, 1998. 111

[9] Basseville, M. and I.V. Nikiforov. *Detection of Abrupt Changes-Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1993. 25, 164

[10] Bauer, D. Order estimation for subspace methods. *Automatica*, 37:1561–1573, 2001. 93, 99

[11] Bauer, D. and L. Ljung. Some facts about the choice of the weighting matrices in larimore type of subspace algorithms. *Automatica*, 38:763–773, 2002. 92

[12] Beard, R.V. Failure Accommodation in Linear Systems throughSelf-Reorganization. Dept. MVT-71-1, Man Vehicle Laboratory, Cambridge, MA., 1971. 25

[13] Bechtler, H., M.W. Browne, P.K. Bansal, and V. Kecman. Neural networks - a new approach to model vapour-compression heat pumps. *International Journal of Energy Research*, 25:591–599, 2001. 60

[14] Burnham, A.J., R. Viveros, and J.F. MacGregor. Frameworks for latent variable multivariate regression. *Journal of Chemometrics*, 10:31–45, 1996. 81, 82, 84

[15] Chen, J. and R.J. Patton. A re-examination of fault detectability and isolability in linear dynamic systems. In *Proc. Safeprocess 94*, pages 590–596, Helsinki, Finland, 1994. 14, 43, 45, 52

[16] Chen, J. and R.J. Patton. *Robust Model-based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publisher, 1999. 106

[17] Chen, J., R.J. Patton, and H. Zhang. Design of unknown input observers and robust fault detection filters. *Int. J. Control*, 63(1):85–105, 1996. 106

[18] Chou, C.T. and M. Verhaegen. Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica*, 33(10):1857–1869, 1997. 147, 157

[19] Chow, E.Y. and A.S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. Auto. Cont.*, AC-29:603–614, 1984. 8, 17, 26, 39, 40, 106, 109, 121, 147, 153, 154

[20] Clark, R.N. Instrument fault detection. *IEEE Trans. Aerospace Electron. Syst.*, 14:456–465, 1978. 26

[21] Clark, R.N. A simplified instrument failure detection scheme. *IEEE Trans. Aerospace Electron. Syst.*, 14:558–563, 1978. 26

[22] Crowe, C.M. Data reconciliation - progress and challenges. *J. Proc. Cont.*, 6:89–98, 1996. 34, 131

[23] Dayal, B. and J.F. MacGregor. Identification of FIR models: methods and robustness issues. *Ind. Engng. Chem. Res.*, 35:4078–4090, 1996. 23

[24] Denham, M. Choosing the number of factors in partial least squares regression: estimating and minimizing the mean squared errors of prediction. *J. of Chemometrics*, 14(4):351–361, 2000. 78

[25] Deyst, J.J. and J.C. Deckert. Rcs jet failure identification for the space shuttle. In *Proc. of IFAC' 1975*, Cambridge, MA, August 1975. 26

[26] Eriksson, L., P. Hagberg, E. Johansson, S. Rännar, O. Whelehan, A. Åström, and T. Lindgren. Multivariate process monitoring of a newsprint mill. application to modelling and predicting cod load resulting from de-inking of recycled paper. *Journal of Chemometrics*, 15(4):337–352, 2001. 60

[27] Madron, F. *Process Plant Performance, Measurement Data Processing for Optimization and Retrofits.* Ellis Horwood, West Sussex, England, 1992. 132

[28] Fadali, M. and H. Emara-Shabaik. Timely robust detection for multirate linear systems. *International J. of Control*, 75:305–313, 2002. 146

[29] Fadali, M. and W. Liu. Fault detection for systems with multirate sampling. In *Proc. of the American Control Conference*, pages 3302–3306, 1998. 146

[30] Fadali, M. and W. Liu. Observer-based robust fault detection for a class of multirate sampled-data linear systems. In *Proc. of the American Control Conference*, pages 97–98, 1999. 146

[31] Frank, P.M. and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *J. Proc. Cont.*, 7(6):403–424, 1997. 106

[32] Frank, P.M. and X. Ding. Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica*, 30(5):789–804, 1994. 106

[33] Frank, P.M. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy — a survey and some new results. *Automatica*, 26:459–474, 1990. 17, 26

[34] Frank, P.M. Enhancement of robustness in observer-based fault detection. *Int. J. Control*, 59:955–981, 1994. 106

[35] Geladi, P. and B.R. Kowalski. Partial least-squares regresion: A tutorial. *Analytica Chimica Acta*, 185:1–17, 1986. 56, 77

[36] Gertler, J. and D. Singer. Augmented models for statistical fault isolation in complex dynamic systems. In *Proceedings of the American Control Conference*, pages 317–322, 1985. 27, 112

[37] Gertler, J. and D. Singer. A new structural framework for parity equation based failure detection and isolation. *Automatica*, 26:381–388, 1990. 27, 112

[38] Gertler, J. Survey of model-based failure detection and isolation in complex plants. *IEEE Cont. Sys. Mag.*, 12:3–11, 1988. 15, 26

[39] Gertler, J. *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker, 1998. 25

[40] Gertler, J. and M.M. Kunwer. Optimal residual decoupling for robust fault diagnosis. *Int. J. Control*, 61:395–421, 1995. 106

[41] Golub, G.H. and C.F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1996. Third Edition. 110, 115, 116

[42] Golub, G.H. Some modified matrix eigenvalue problems. *SIAM Rev.*, 15:318–334, 1973. 90, 158

[43] Gudi, R., S.L. Shah, and M. Gary. Adaptive multirate state and parameter estimation strategies with application to a bioreactor. *AIChE J.*, 41:2451–2464, 1994. 146

[44] Gustafson, D.E., A.S. Willsky, and J.Y. Wang. Cardiac Arrhythmia Detection and Classification through Signal Analysis. Report of The Charles Stark Draper Laboratory, MIT, Cambridge, MA., 1975. 26

[45] Gustafsson, F. and S.F. Graebe. Closed-loop performance monitoring in the presence of system changes and disturbances. *Automatica*, 34(11):1311–1326, 1998. 13, 106, 108

[46] Gustafsson, T. and B. Rao. Statistical analysis of subspace-based estimation of reducedrank linear regressions. *IEEE Transactions on Signal Processing*, 50:151–159, 2002. 103

[47] Hamelin, F. and D. Sauter. Robust fault detection in uncertain dynamic systems. *Automatica*, 36:1747–1754, 2000. 106

[48] Haykin, S. *Neural Networks A Comprehensive Foundation*. Prentice-Hall, 1994. 60

[49] Himmelblau, D.M. *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*. Elsevier, New York, 1978. 25

[50] Hocking, R.R. The analysis and selection of variables in linear regression. *Biometrics*, 32:1–49, 1976. 62

[51] Hoskuldsson, A. PLS regression methods. *Journal of Chemometrics*, 2:211–228, 1988. 77, 82

[52] Hotelling, H. The most predicable criterion. *Journal of Educational Psychology*, 26:139–142, 1935. 56

[53] Hotelling, H. Relations between two sets of variables. *Biomatrika*, 28:321–377, 1936. 56

[54] Isermann, R. Process fault detection based on modeling and estimation methods–a survey. *Automatica*, 20:387–404, 1984. 26

[55] Jackson, J.E. and G. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21:341–349, 1979. 33

[56] Jackson, J.E. *A User's Guide to Principal Components*. Wiley-Interscience, New York, 1991. 23, 30

[57] Johnson, C.D. *Process control instrumentation technology*. John Wiley and Sons Inc., 3rd edition, 1988. 138

[58] Johnson, R.A and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Inc., 1998. Fourth Edition. 84, 85, 111, 154, 160, 164

[59] Jones, H.L. Failure Detection in Linear Systems. Ph.D. Thesis, MIT, Cambridge, MA., 1973. 25

[60] Kaspar, M.H. and W.H. Ray. Chemometric methods for process monitoring and high-performance controller design. *AIChE J.*, 38(10):1593–1608, October 1992. 79, 82

[61] Keller, J.Y. Fault isolation filter design for linear stochastic systems. *Automatica*, 35:1701–1706, 1995. 28, 30

[62] Khargonekar, P., K. Poola, and A. Tannenbaum. Robust control of linear time-invariant plants using periodic compensation. *IEEE Trans. Auto. Cont.*, 30:1088–1096, 1985. 146

[63] Kresta, J.V., J.F. MacGregor, and T.E. Marlin. Multivariate statistical monitoring of processes. *Can. J. Chem. Eng.*, 69(1):35–47, 1991. 30

[64] Kuss, M. and T. Graepel. The geometry of kernel canonical correlation analysis. Technical Report 108, Max Planck Institute for Biological Cybernetics, 2003. 82

[65] Lakshminarayanan, S., S.L. Shah, and K. Nandakumar. Modeling and control of multivariable process: dynamic PLS approach. *AIChE J.*, 43:2307–2322, 1997. 23

[66] Larimore, W.E. System identification, reduced-order filtering and modeling via canonical variate analysis. In H. S. Rao and T. Torato, editors, *Proceedings of the 1983 American Control Conference*, pages Vol. 1, 445–451, New York, 1983. IEEE. 82

[67] Larimore, W.E. Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of the 29th Conference on Decision and Control*, pages 635–639, Honolulu, Hawaii, 1990. 20, 82

[68] Li, D., S.L. Shah, T. Chen, and K. Qi. Application of dual-rate modeling to CCR octane quality inferential control. In *Proceedings of the IFAC symposium dynamics and control of process systems*, pages 417–421, Cheju, Korea, 2001. 76, 147

[69] Li, W. and S.L. Shah. Fault detection and isolation in non-uniformly sampled systems. In *accepted by IFAC DYCOPS 7*, Boston, MA, USA, July 2004. 146

[70] Li, W. and S.L. Shah. Structured residual vector-based approach to sensor fault detection and isolation. *J. Proc. Cont.*, 12:429–443, 2002. 26, 27, 28, 106, 110, 112, 113, 153, 163

[71] Liu, B. and J. Si. Fault isolation filter design for time-invariant systems. *IEEE Trans. Auto. Cont.*, 42:704–707, 1997. 29

[72] Ljung, L. *System identification: Theory for the user.* Prentice-Hall Inc., 2nd edition, 1999. 15, 18, 19, 20, 56

[73] Lopes, J.A., J.C. Menezes, J.A. Westerhuis, and A.K. Smilde. Multiblock pls analysis of an industrial pharmaceutical process. *Journal of Chemometrics*, 80(4):419–427, 2002. 60

[74] Lou, X., A.S. Willsky, and G.C. Verghese. Optimally robust redundancy relations for failure detection in uncertain systems. *Automatica*, 22(3):333–344, 1986. 106

[75] MacGregor, J.F., T. Kourti, and J. Kresta. Multivariate identification: a study of several methods. In *IFAC Symp. ADCHEM-91*, Toulouse, 1991. 23

[76] MacGregor, J.F. and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3:403–414, 1995. 33

[77] Madron, F. A new approach to the identification of gross errors in chemical engineering measurements. *Chem. Eng. Sci.*, pages 1855–1860, 1985. 34

[78] Mah, R.S.H., G.M. Stanley, and D. Downing. Reconciliation and rectification of process flow and inventory data. *Ind. Eng. Chem. Proc. Des. Dev.*, 15:175, 1976. 34, 35

[79] Mah, R.S.H. *Chemical process strunctures and information flows.* Chem. Eng. Ser. Butterworth, Boston, 1990. 34, 132

[80] Mah, R.S.H. and A.C. Tamhane. Detection of gross errors in process data. *AIChE J.*, 28:828–830, 1982. 35

[81] Mangoubi, R.S. *Robust Estimation and Failure Detection.* Springer-Verlag, 1998. 25

[82] Manne, R. Analysis of two partial-least-squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 2:283–290, 1987. 77

[83] Marlin, T.E. *Process Control - Designing Processes and Control Systems for Dynamic Performance.* McGraw-Hill Inc., New York, 1995. 98, 118, 135

[84] Mehra, R.K. and I. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637–640, 1971. 25

[85] Miller, P., R.E. Swanson, and C.F. Heckler. Contribution plots: the missing link in multivariate quality control. In *Fall Conf. Of the ASQC and ASA*, 1993. Milwaukee, WI. 33

[86] Moonen, M., B. DeMoor, L. Vandenberghe, and J. Vandewalle. On and off-line identification of linear state-space models. *International Journal of Control*, 49:219–232, 1989. 147

[87] Næs, T. Leverage and influence measures for principal component regression. *Chemometrics and Intelligent Laboratory Systems*, 5:155–168, 1988. 56

[88] Narasimhan, S. and R.S.H. Mah. Generalized likelihood ratio method for gross error identification. *AIChE J.*, 33:1514–1521, 1987. 35

[89] Negiz, A. and A. Cinar. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE J.*, 43(8):2002–2020, 1997. 82

[90] Nimmo, I. Adequately address abnormal situation operations. *Chem. Eng. Prog.*, 91:36–45, 1995. 2

[91] Nogita, S. Statistical test and adjustment of process data. *Ind. Eng. Chem. Proc. Des. Dev.*, 11:197–200, 1972. 34

[92] Nyberg, M. and L. Nielsen. A universal chow-willsky scheme and detectability criteria. *IEEE Trans. Auto. Cont.*, 45:152–156, 2000. 38

[93] Nyberg, M. and L. Nielson. Parity functions as universal residual generators and tool for fault detectability analysis. In *IEEE Conf. on Decision and Control*, pages 4483–4489, San Diego, California, 1997. 39, 43, 45

[94] Oshima, M., I. Hashimoto, M. Takeda, M. Yoneyama, and F. Goto. Multirate multivariate model prediction control and its application to a semi-commercial polymerization reactor. In *Proc. of the American Control Conference*, pages 1576–1581, 1992. 146

[95] Van Overschee, P. and B. De Moor. N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994. 20

[96] Van Overschee, P. and B. De Moor. N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994. 147

[97] Van Overschee, P. and B. De Moor. A unifying theorem for three subspace system identification algorithms. *Automatica*, 31:1853–1864, 1995. 147

[98] Van Overschee, P. and B. De Moor. *Subspace Identification for Linear Systems-Theory, Implementation, Applications.* Kluwer Academic Publisher, Boston, MA, 1996. 22, 56

[99] Patton, R., P.M. Frank, and R. Clark. *Fault Diagnosis in Dynamic Systems.* Prentice Hall, 1989. 25

[100] Patton, R. and J. Chen. Robust fault detection of jet engine sensor systems using eigenstructure assignment. *Journal of Guidance, Control, and Dynamics*, 15:1491–1497, 1992. 106

[101] Patton, R. and J. Chen. On eigenstructure assisgnment for robust fault diagnosis. *International Journal of Robust and Nonlinear control*, 10:1193–1208, 2000. 106

[102] Patton, R., P.M. Frank, and R. Clark (Eds). *Issues of Fault Diagnosis for Dynamic Systems.* Springer-Verlag Ltd., 2000. 25

[103] Pearson, K. On line and planes of closest fit to systems of points in space. *Phil. Mag.*, 2:559, 1901. 30

[104] Proakis, J.G. *Digital Communications.* McGraw-Hill Book Company, 2nd edition, 1989. 116

[105] Qin, S.J. and W. Li. Detection, identification, and reconstruction of faulty sensors with maximized sensitivity. *AIChE Journal*, 45(9):1963–1976, 1999. 26, 27, 121

[106] Qin, S.J. and W. Li. Detection an identification of faulty sensors in dynamic processes. *AIChE J.*, 47:1581–1593, 2001. 106

[107] Rao, C. The use and interpretation of principal component analysis in applied research. *The India J. of Statistics, Series A*, 26:329–358, 1964. 158

[108] Reilly, P.M. and R.E. Carpani. Application of statistical theory of adjustment to material balances. In *13th Canadian Chem. Eng. Conference*, Ottawa, Canada, 1963. 35

[109] Riggs, J.B. *Chemical process control.* Ferret Publishing, 1999. 135

[110] Ripps, D.L. Adjustment of experimental data. *Chem. Eng. Prog. Symp. Ser.*, 61:8–13, 1965. 34

[111] Romagnoli, J.A. On data reconciliation constraints processing and treatment of bias. *Chem. Eng. Sci.*, 38:1107–1117, 1983. 34

[112] Romagnoli, J.A. and G. Stephanopoulos. Rectification of process measurement data in the presence of gross errors. *Chem. Eng. Sci.*, 36(11):1849–1863, 1981. 34

[113] Di Ruscio, D. A weighted view on the partial least-squares algorithm. *Automatica*, 36:831–850, 2000. 83, 88, 89, 95

[114] Narasimhan, S. and C. Jordache. *Data Reconciliation and Gross Error Detection. An Intelligent Use of Process Data.* Gulf Publishing Company, 2000. 131, 132

[115] Saberi, A., A. Stoorvogel, P. Sannuti, and H. Nieman:1997. Fundamental problems in fault detection and identification. In *Proceedings of the 38th Conference on Decision and Control*, pages 3108–3113. IEEE, 1999. 38

[116] Sánchez, M. and J. Romagnoli. *Processing and Reconciliation for Chemical Process Operations.* Academic Press, 2000. 132

[117] Schaper, C., W. Larimore, D. Seborg, and D. Mellichamp. Identification of chemical processes using canonical variate analysis. *Computers Chem. Engng*, 1994:55–69, 1994. 82

[118] Shen, L. and P. Hsu. Robust design of fault isolation observers. *Automatica*, 34:1421–1429, 1998. 106

[119] Sheng, J., T. Chen, and S.L. Shah. Generalized predictive control for non-uniformly sampled systems. *J. of Process Control*, 12:875–885, 2002. 146, 147, 148, 150, 152

[120] Shi, R. and J.F. MacGregor. Modeling of dynamic systems using latent variable and subspace methods. *Journal of Chemometrics*, 14:423–439, 2000. 23

[121] Söderström, T. and P. Stoica. *Instrumental variable methods for system identification.* Springer-Verlag, 1983. 56

[122] Söderström, T., U.Soverini, and K. Mahata. Perspectives on errors-in-variables estimation for dynamic systems. *Signal processing*, 82(8):1139–1154, 2002. 13

[123] Tamhane, A.C. A note on the use of residuals for detecting an outlier in linear regression. *Biometrika*, 69:488–499, 1982. 35

[124] Tracy, N.D., J.C. Young, and R.L. Mason. Multivariate control charts for individual observations. *Journal of Quality Technology*, 24:88–95, 1992. 32

[125] Venkatasubramanian, V. Process fault detectiona and diagnosis: past, present and future. In *Proc. of IFAC On-line fault detection and supervision in the chemical process industries*, Jejudo Island, Korean, July 2001. 2

[126] Verhaegen, M. and P. Dewilde. Subspace model identification. part i: the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210, 1992. 20

[127] Willsky, A.S. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976. 26

[128] Wise, B.M. and N.B. Gallagher. The process chemometrics approach to process monitoring and fault detection. *Int. J. Control*, 6(6):329–348, 1996. 82

[129] Wise, B.M., N.B. Gallagher, and J.F. MacGregor. The process chemometrics approach to process monitoring and fault detection. In *Proceedings of the IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries*, 1995. 26

[130] Wise, B.M. and N.L. Ricker. Recent advances in multivariate statistical process control: Improving robustness and sensitivity. In *Proceedings of the IFAC. ADCHEM Symposium*, pages 125–130, 1991. 30

[131] Wold, H. *Multivariate Analysis.* Academic, 1966. 30

[132] Wold, S. Cross validatory estimation of the number of components in factor and principal component analysis. *Technometrics*, 20(4):397–406, 1978. 78

[133] Wold, S., N. Kettaneh-Wold, and B. Skagerberg. Nonlinear PLS modelling. *Chemometrics Intell. Lab. Syst.*, 7:53–65, 1989. 23

[134] Yoon, S. and J.F. MacGregor. Fault diagnosis with multivariate statistical models part I: using steady state fault signatures. *J. of Process Control*, 11:387–400, 2001. 30, 118

[135] Zhang, P., S. Ding, G. Wang, and D. Zhou. Fault detection for multirate sample-data systems with time delay. *International J. of Control*, 75:1457–1471, 2002. 146

[136] Zhong, M., S.X. Ding, J. Lam, and H. Wang. An LMI approach to design robust fault detection filter for uncertain LTI systems. *Automatica*, 39(3):543–550, 2003. 106, 107